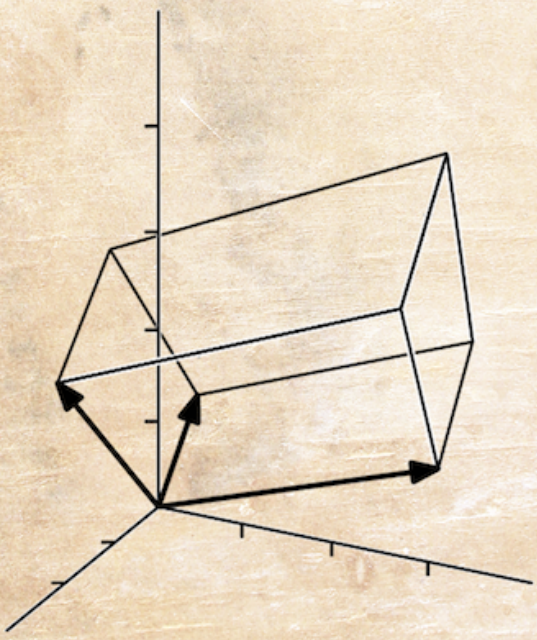
NO BULLSHIT

guide to

LINEAR ALGEBRA



by Ivan Savov

www.ebook3000.com

www.ebook3000.com

No bullshit guide to linear algebra

Ivan Savov

April 30, 2016

www.ebook3000.com

No bullshit guide to linear algebra
by Ivan Savov

Copyright © Ivan Savov, 2015, 2016. All rights reserved.

Published by Minireference Co.
Montréal, Québec, Canada
minireference.com | @minireference | fb.me/noBSguide
For inquiries, contact the author at ivan.savov@gmail.com

Near-final release
v0.91 hg changeset: 314:578e9ac33e27

ISBN 978-0-9920010-1-8
10 9 8 7 6 5 4 3 2 1

www.ebook3000.com

Contents

Preface **vii**

Introduction **1**

1 Math fundamentals **9**
1.1 Solving equations 10
1.2 Numbers 12
1.3 Variables 16
1.4 Functions and their inverses 18
1.5 Basic rules of algebra 21
1.6 Solving quadratic equations 25
1.7 The Cartesian plane 29
1.8 Functions 32
1.9 Function reference 38
1.10 Polynomials 54
1.11 Trigonometry 58
1.12 Trigonometric identities 63
1.13 Geometry 65
1.14 Circle 67
1.15 Solving systems of linear equations 69
1.16 Set notation 73
1.17 Math problems 79

2 Vectors **89**
2.1 Vectors 90
2.2 Basis 98
2.3 Vector products 99
2.4 Complex numbers 102
2.5 Vectors problems 107

3 Intro to linear algebra **111**
3.1 Introduction 111
3.2 Review of vector operations 117

www.ebook3000.com

3.3 Matrix operations . . . 121
3.4 Linearity . . . 126
3.5 Overview of linear algebra . . . 131
3.6 Introductory problems . . . 135

4 Computational linear algebra 137
4.1 Reduced row echelon form . . . 138
4.2 Matrix equations . . . 150
4.3 Matrix multiplication . . . 154
4.4 Determinants . . . 158
4.5 Matrix inverse . . . 169
4.6 Computational problems . . . 176

5 Geometrical aspects of linear algebra 181
5.1 Lines and planes . . . 181
5.2 Projections . . . 189
5.3 Coordinate projections . . . 194
5.4 Vector spaces . . . 199
5.5 Vector space techniques . . . 210
5.6 Geometrical problems . . . 220

6 Linear transformations 223
6.1 Linear transformations . . . 223
6.2 Finding matrix representations . . . 234
6.3 Change of basis for matrices . . . 245
6.4 Invertible matrix theorem . . . 249
6.5 Linear transformations problems . . . 256

7 Theoretical linear algebra 257
7.1 Eigenvalues and eigenvectors . . . 258
7.2 Special types of matrices . . . 271
7.3 Abstract vector spaces . . . 277
7.4 Abstract inner product spaces . . . 281
7.5 Gram–Schmidt orthogonalization . . . 288
7.6 Matrix decompositions . . . 292
7.7 Linear algebra with complex numbers . . . 298
7.8 Theory problems . . . 313

8 Applications 317
8.1 Balancing chemical equations . . . 318
8.2 Input–output models in economics . . . 320
8.3 Electric circuits . . . 321
8.4 Graphs . . . 327
8.5 Fibonacci sequence . . . 330
8.6 Linear programming . . . 332

www.ebook3000.com

8.7 Least squares approximate solutions . . . 333
8.8 Computer graphics . . . 342
8.9 Cryptography . . . 354
8.10 Error correcting codes . . . 366
8.11 Fourier analysis . . . 375
8.12 Applications problems . . . 389

9 Probability theory **391**
9.1 Probability distributions . . . 391
9.2 Markov chains . . . 398
9.3 Google's PageRank algorithm . . . 404
9.4 Probability problems . . . 410

10 Quantum mechanics **411**
10.1 Introduction . . . 412
10.2 Polarizing lenses experiment . . . 418
10.3 Dirac notation for vectors . . . 425
10.4 Quantum information processing . . . 431
10.5 Postulates of quantum mechanics . . . 434
10.6 Polarizing lenses experiment revisited . . . 448
10.7 Quantum physics is not that weird . . . 452
10.8 Quantum mechanics applications . . . 457
10.9 Quantum mechanics problems . . . 473

End matter **475**
Conclusion . . . 475
Social stuff . . . 477
Acknowledgements . . . 477
General linear algebra links . . . 477

A Answers and solutions **479**

B Notation **499**
Math notation . . . 499
Set notation . . . 500
Vectors notation . . . 500
Complex numbers notation . . . 501
Vector space notation . . . 501
Notation for matrices and matrix operations . . . 502
Notation for linear transformations . . . 503
Matrix decompositions . . . 503
Abstract vector space notation . . . 504

www.ebook3000.com

Concept maps

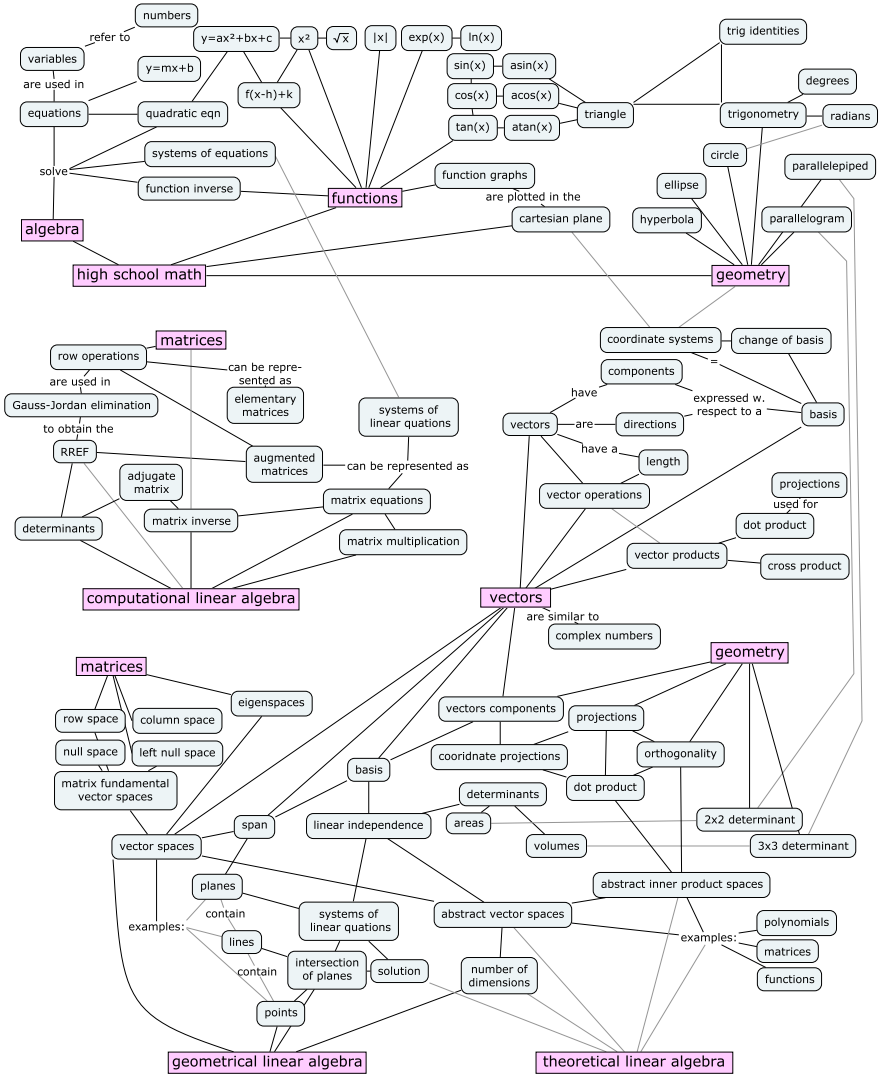


Figure 1: This concept map illustrates the prerequisite topics of high school math covered in Chapter 1 and vectors covered in Chapter 2. Also shown are the topics of computational and geometrical linear algebra covered in Chapters 4 and 5.

www.ebook3000.com

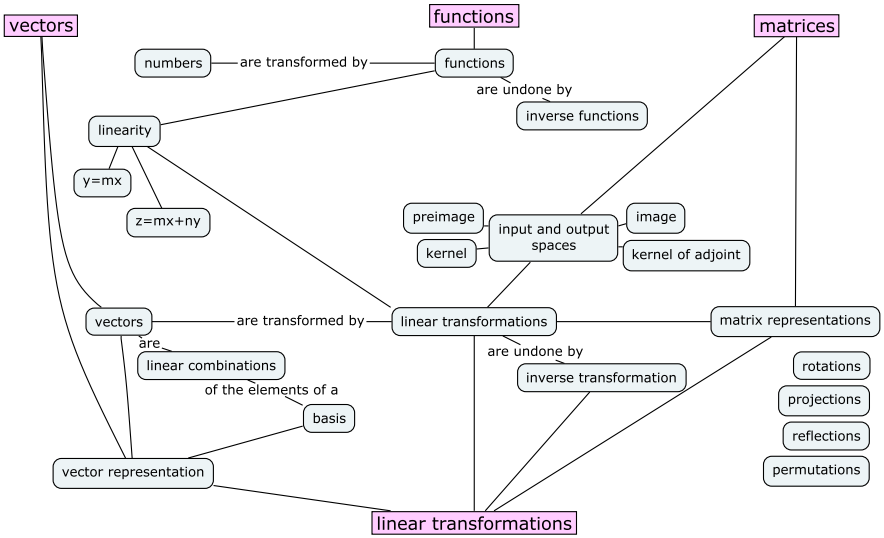


Figure 2: Chapter 6 covers linear transformations and their properties.

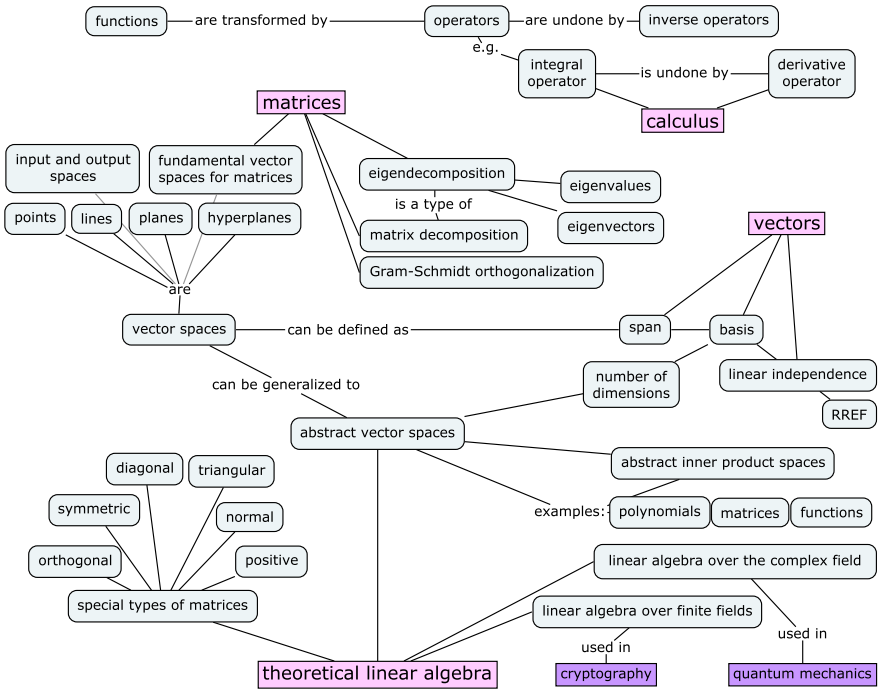


Figure 3: Chapter 7 covers theoretical aspects of linear algebra.

www.ebook3000.com

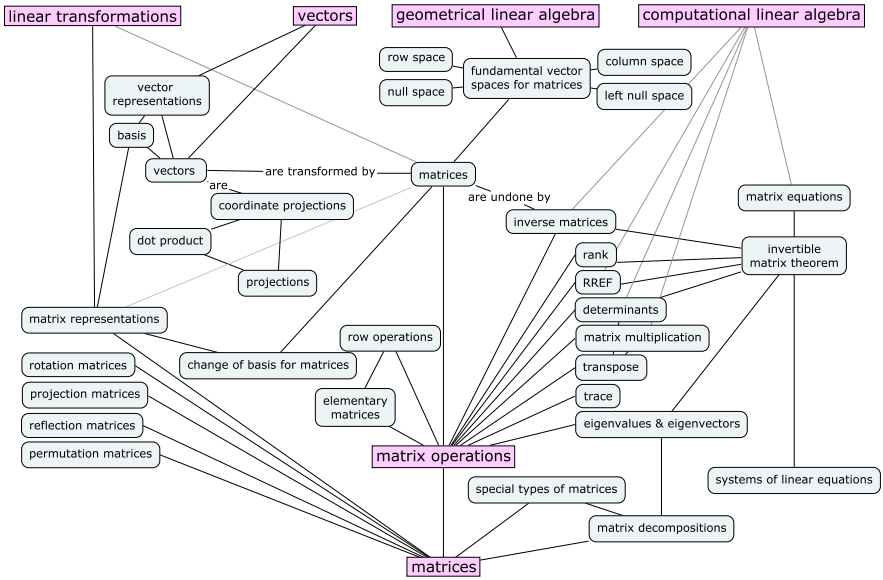


Figure 4: Matrix operations and matrix computations play an important role throughout this book. Matrices are used to implement linear transformations, systems of linear equations, and various geometrical computations.

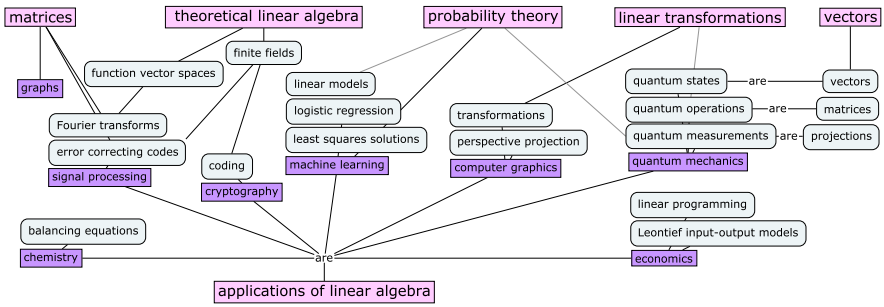


Figure 5: The book concludes with three chapters on linear algebra applications. In Chapter 8 we'll discuss applications to science, economics, business, computing, and signal processing. Chapter 9 on probability theory and Chapter 10 on quantum mechanics serve as examples of advanced subjects that you can access once you learn linear algebra.

www.ebook3000.com

Preface

This book is about linear algebra and its applications. The material is covered at the level of a first-year university course with more advanced concepts also being presented. The book is written in a clean, approachable style that gets to the point. Both practical and theoretical aspects of linear algebra are discussed, with extra emphasis on explaining the connections between concepts and building on material students are already familiar with.

Since it includes all necessary prerequisites, this book is suitable for readers who don't feel "comfortable" with fundamental math concepts, having never learned them well, or having forgotten them over the years. The goal of this book is to **give access to advanced mathematical modelling tools** to everyone interested in learning, regardless of their academic background.

Why learn linear algebra?

Linear algebra is one of the most useful undergraduate math subjects. The practical skills like manipulating vectors and matrices that students learn will come in handy for physics, computer science, statistics, machine learning, and many other areas of science. Linear algebra is essential for anyone pursuing studies in science.

In addition to being useful, learning linear algebra can also be a lot of fun. Readers will experience *knowledge buzz* from understanding the connections between concepts and seeing how they fit together. Linear algebra is one of the most fundamental subjects in mathematics and it's not uncommon to experience mind-expanding moments while studying this subject.

The powerful concepts and tools of linear algebra form a bridge toward more advanced areas of mathematics. For example, learning about *abstract vector spaces* will help students recognize the common "vector structure" in seemingly unrelated mathematical objects like matrices, polynomials, and functions. Linear algebra techniques can be applied not only to standard vectors, but to *all* mathematical

objects that are vector-like!

What's in this book?

Each section in this book is a self-contained tutorial that covers the definitions, formulas, and explanations associated with a single topic. Consult the concept maps on the preceding pages to see the topics covered in the book and the connections between them.

The book begins with a review chapter on numbers, algebra, equations, functions, and trigonometry (Chapter 1) and a review chapter on vectors (Chapter 2). Anyone who hasn't seen these concepts before, or who feels their math and vector skills are a little "rusty" should read these chapters and work through the exercises and problems provided. Readers who feel confident in their high school math abilities can jump straight to Chapter 3 where the linear algebra begins.

Chapters 4 through 7 cover the core topics of linear algebra: vectors, bases, analytical geometry, matrices, linear transformations, matrix representations, vector spaces, inner product spaces, eigenvectors, and matrix decompositions.

Chapters 8, 9, and 10 discuss various applications of linear algebra. Though not likely to appear on any linear algebra final exam, these chapters serve to demonstrate the power of linear algebra techniques and their relevance to many areas of science. The mini-course on quantum mechanics (Chapter 10) is unique to this book.

Is this book for you?

The quick pace and lively explanations in this book provide interesting reading for students and non-students alike. Whether you're learning linear algebra for a course, reviewing material as a prerequisite for more advanced topics, or generally curious about the subject, this guide will help you find your way in the land of linear algebra. The short-tutorial format cuts to the chase: we're all busy adults with no time to waste!

This book can be used as the main textbook for any university-level linear algebra course. It contains everything students need to know to prepare for a linear algebra final exam. Don't be fooled by the book's small format: it's all in here. The text is compact because it distills the essentials and removes the unnecessary cruft.

Publisher

The genesis of the NO BULLSHIT GUIDE textbook series dates back to my student days, when I was required to purchase expensive course textbooks, which were long and tedious to read. I said to myself

that "something must be done," and started a textbook company to produce textbooks that explain math and physics concepts clearly, concisely, and affordably.

The goal of `Minireference Publishing` is to fix the first-year science textbooks problem. Mainstream textbooks suck, so we're doing something about it. We want to set the bar higher and redefine readers' expectations for what a textbook should be! Using print-on-demand and digital distribution strategies allows us to provide readers with high quality textbooks at reasonable prices.

About the author

I have been teaching math and physics for more than 15 years as a private tutor. Through this experience, I learned to explain difficult concepts by breaking complicated ideas into smaller chunks. An interesting feedback loop occurs when students learn concepts in small chunks: the *knowledge buzz* they experience when concepts "click" into place motivates them to continue learning more. I know this from first-hand experience, both as a teacher and as a student. I completed my undergraduate studies in electrical engineering, then stayed on to earn a M.Sc. in physics, and a Ph.D. in computer science.

Linear algebra played a central role throughout my studies. With this book, I want to share with you some of what I've learned about this expansive subject.

Ivan Savov
Montreal, 2016

Introduction

There have been countless advances in science and technology in recent years. Modern science and engineering fields have developed advanced models for understanding the real world, predicting the outcomes of experiments, and building useful technology. We're still far from obtaining a theory of everything that can predict the future, but we understand a lot about the natural world at many levels of description: physical, chemical, biological, ecological, psychological, and social. Anyone interested in being part of scientific and technological advances has no choice but to learn mathematics, since mathematical models are used throughout all fields of study. The linear algebra techniques you'll learn in this book are some of the most powerful mathematical modelling tools that exist.

At the core of linear algebra lies a very simple idea: *linearity*. A function f is *linear* if it obeys the equation

$$f(a\mathbf{x}_1 + b\mathbf{x}_2) = af(\mathbf{x}_1) + bf(\mathbf{x}_2),$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are any two inputs suitable for the function. We use the term *linear combination* to describe any expression constructed from a set of variables by multiplying each variable by a constant and adding the results. In the above equation, the linear combination $a\mathbf{x}_1 + b\mathbf{x}_2$ of the inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ is transformed into the linear combination $af(\mathbf{x}_1) + bf(\mathbf{x}_2)$ of the outputs of the function $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$. **Linear functions transform linear combinations of their inputs into the *same* linear combination of their outputs.** That's it, that's all! Now you know everything there is to know about linear algebra. The rest of the book is just details.

A significant proportion of the models used by scientists and engineers describe *linear relationships* between quantities. Scientists, engineers, statisticians, business folk, and politicians develop and use linear models to make sense of the systems they study. In fact, linear models are often used to model even nonlinear (more complicated) phenomena. There are several good reasons for using linear models. The first reason is that linear models are very good at *approximating*

the real world. Linear models for nonlinear phenomena are referred to as *linear approximations*. If you've previously studied calculus, you'll remember learning about *tangent lines*. The tangent line to a curve $f(x)$ at x_o is given by the equation

$$T(x) = f'(x_o)\big(x - x_o\big) + f(x_o).$$

This line has slope $f'(x_o)$ and passes through the point $(x_o, f(x_o))$. The equation of the tangent line $T(x)$ serves to approximate the function $f(x)$ near x_o. Using linear algebra techniques to model nonlinear phenomena can be understood as a multivariable generalization of this idea.

Linear models can also be combined with nonlinear transformations of the model's inputs or outputs to describe nonlinear phenomena. These techniques are often employed in machine learning: *kernel methods* are arbitrary non-linear transformations of the inputs of a linear model, and the *sigmoid activation curve* is used to transform a smoothly-varying output of a linear model into a hard `yes` or `no` decision.

Perhaps the main reason linear models are widely used is because they are easy to describe mathematically, and easy to "fit" to real-world systems. We can obtain the parameters of a linear model for a real-world system by analyzing its behaviour for relatively few inputs. We'll illustrate this important point with an example.

Example At an art event, you enter a room with a multimedia setup. A drawing canvas on a tablet computer is projected on a giant screen. Anything you draw on the tablet will instantly appear projected on the giant screen. The user interface on the tablet screen doesn't give any indication about how to hold the tablet "right side up." What is the fastest way to find the correct orientation of the tablet so your drawing will not appear rotated or upside-down?

This situation is directly analogous to the tasks scientists face every day when trying to model real-world systems. The canvas on the tablet describes a two-dimensional *input space*, and the wall projection is a two-dimensional *output space*. We're looking for the unknown transformation T that maps the pixels of the tablet screen (the input space) to coloured dots on the wall (the output space). If the unknown transformation T is a linear transformation, we can learn its parameters very quickly.

Let's describe each pixel in the input space with a pair of coordinates (x, y) and each point on the wall with another pair of coordinates (x', y'). The unknown transformation T describes the mapping of pixel coordinates to wall coordinates:

$$(x, y) \xrightarrow{T} (x', y').$$

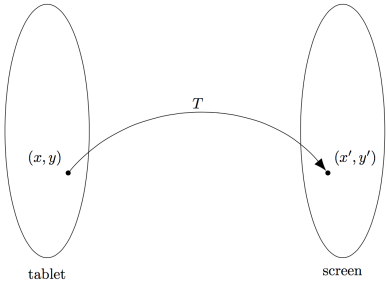


Figure 6: An unknown linear transformation T maps "tablet coordinates" to "screen coordinates." How can we characterize T?

To uncover how T transforms (x, y)-coordinates to (x', y')-coordinates, you can use the following three-step procedure. First put a dot in the lower left corner of the tablet to represent the *origin* $(0, 0)$ of the xy-coordinate system. Observe the location where the dot appears on the wall—we'll call this location the origin of the $x'y'$-coordinate system. Next, make a short horizontal swipe on the screen to represent the x-direction $(1, 0)$ and observe the transformed $T(1, 0)$ that appears on the wall. As the final step, make a vertical swipe in the y-direction $(0, 1)$ and see the transformed $T(0, 1)$ that appears on the wall. By noting how the xy-coordinate system is mapped to the $x'y'$-coordinate system, you can determine which orientation you must hold the tablet for your drawing to appear upright when projected on the wall. **Knowing the outputs of a linear transformation T for all "directions" in its inputs space allows us to completely characterize T.**

In the case of the multimedia setup at the art event, we're looking for an unknown transformation T from a two-dimensional input space to a two-dimensional output space. Since T is a linear transformation, it's possible to completely describe T with only two swipes. Let's look at the math to see why this is true. Can you predict what will appear on the wall if you make an angled swipe in the $(2, 3)$-direction? Observe that the point $(2, 3)$ in the input space can be obtained by moving 2 units in the x-direction and 3 units in the y-direction: $(2, 3) = (2, 0) + (0, 3) = 2(1, 0) + 3(0, 1)$. Using the fact that T is a linear transformation, we can predict the output of the transformation when the input is $(2, 3)$:

$$T(2, 3) = T(2(1, 0) + 3(0, 1)) = 2T(1, 0) + 3T(0, 1).$$

The projection of the diagonal swipe in the $(2, 3)$-direction will have a length equal to 2 times the unit x-direction output $T(1, 0)$ plus 3 times the unit y-direction output $T(0, 1)$. Knowledge of the outputs

of the two swipes $T(1,0)$ and $T(0,1)$ is sufficient to determine the linear transformation's output for any input (a,b). Any input (a,b) can be expressed as a linear combination: $(a,b) = a(1,0) + b(0,1)$. The corresponding output will be $T(a,b) = aT(1,0) + bT(0,1)$. Since we know $T(1,0)$ and $T(0,1)$, we can calculate $T(a,b)$.

TL;DR Linearity allows us to analyze multidimensional processes and transformations by studying their effects on a small set of inputs. This is the essential reason linear models are so prominent in science. Probing a linear system with each "input direction" is enough to completely characterize the system. Without this linear structure, characterizing unknown input-output systems is a much harder task. Linear algebra is the study of linear structure, in all its details. The theoretical results and computational procedures of you'll learn apply to all things linear and vector-like.

Linear transformations

You can think of linear transformations as "vector functions" and understand their properties in analogy with the properties of the regular functions you're familiar with. The action of a function on a number is similar to the action of a linear transformation on a vector:

$$\begin{aligned}
\text{function } f : \mathbb{R} \to \mathbb{R} &\quad\Leftrightarrow\quad \text{linear transformation } T : \mathbb{R}^n \to \mathbb{R}^m \\
\text{input } x \in \mathbb{R} &\quad\Leftrightarrow\quad \text{input } \vec{x} \in \mathbb{R}^n \\
\text{output } f(x) \in \mathbb{R} &\quad\Leftrightarrow\quad \text{output } T(\vec{x}) \in \mathbb{R}^m \\
\text{inverse function } f^{-1} &\quad\Leftrightarrow\quad \text{inverse transformation } T^{-1} \\
\text{zeros of } f &\quad\Leftrightarrow\quad \text{kernel of } T
\end{aligned}$$

Studying linear algebra will expose you to many topics associated with linear transformations. You'll learn about concepts like vector spaces, projections, and orthogonalization procedures. Indeed, a first linear algebra course introduces many advanced, abstract ideas; yet all the new ideas you'll encounter can be seen as extensions of ideas you're already familiar with. Linear algebra is the vector-upgrade to your high-school knowledge of functions.

Prerequisites

To understand linear algebra, you must have some preliminary knowledge of fundamental math concepts like numbers, equations, and functions. For example, you should be able to tell me the meaning of the parameters m and b in the equation $f(x) = mx + b$. If you do not

feel confident about your basic math skills, don't worry. Chapter 1 is specially designed to help bring you quickly up to speed on the material of high school math.

It's not a requirement, but it helps if you've previously used vectors in physics. If you haven't taken a mechanics course where you saw velocities and forces represented as vectors, you should read Chapter 2, as it provides a short summary of vectors concepts usually taught in the first week of Physics 101. The last section in the vectors chapter (Section 2.4) is about complex numbers. You should read that section at some point because we'll use complex numbers in Section 7.7 later in the book.

Executive summary

The book is organized into ten chapters. Chapters 3 through 7 are the core of linear algebra. Chapters 8 through 10 contain "optional reading" about linear algebra applications. The concept maps on pages iv, v, and vi illustrate the connections between the topics we'll cover. I know the maps are teeming with concepts, but don't worry—the book is split into tiny chunks, and we'll navigate the material step by step. It will be like Mario World, but in n dimensions and with a lot of bonus levels.

Chapter 3 is an introduction to the subject of linear algebra. Linear algebra is the math of vectors and matrices, so we'll start by defining the mathematical operations we can perform on vectors and matrices.

In Chapter 4, we'll tackle the computational aspects of linear algebra. By the end of this course, you will know how to solve systems of equations, transform a matrix into its reduced row echelon form, compute the product of two matrices, and find the determinant and the inverse of a square matrix. Each of these computational tasks can be tedious to carry out by hand and can require lots of steps. There is no way around this; we must do the grunt work before we get to the cool stuff.

In Chapter 5, we'll review the properties and equations of basic geometrical objects like points, lines, and planes. We'll learn how to compute projections onto vectors, projections onto planes, and distances between objects. We'll also review the meaning of vector coordinates, which are lengths measured with respect to a basis. We'll learn about linear combinations of vectors, the span of a set of vectors, and formally define what a vector space is. In Section 5.5 we'll learn how to use the *reduced row echelon form* of a matrix, in order to describe the fundamental spaces associated with the matrix.

Chapter 6 is about linear transformations. Armed with the com-

putational tools from Chapter 4 and the geometrical intuition from Chapter 5, we can tackle the core subject of linear algebra: linear transformations. We'll explore in detail the correspondence between linear transformations (vectors functions $T : \mathbb{R}^n \to \mathbb{R}^m$) and their representation as $m \times n$ matrices. We'll also learn how the coefficients in a matrix representation depend on the choice of basis for the input and output spaces of the transformation. Section 6.4 on the **invertible matrix theorem** serves as a midway checkpoint for your understanding of linear algebra. This theorem connects several seemingly disparate concepts: reduced row echelon forms, matrix inverses, row spaces, column spaces, and determinants. The **invertible matrix theorem** links all these concepts and highlights the properties of invertible linear transformations that distinguish them from non-linear transformations. Invertible transformations are one-to-one correspondences (bijections) between the vectors in the input space and the vectors in the output space.

Chapter 7 covers more advanced theoretical topics of linear algebra. We'll define the eigenvalues and the eigenvectors of a square matrix. We'll see how the eigenvalues of a matrix tell us important information about the properties of the matrix. We'll learn about some special names given to different types of matrices, based on the properties of their eigenvalues. In Section 7.3 we'll learn about abstract vector spaces. Abstract vectors are mathematical object that—like vectors—have components and can be scaled, added, and subtracted component-wise. Section 7.7 will discuss linear algebra with complex numbers. Instead of working with vectors with real coefficients, we can do linear algebra with vectors that have complex coefficients. This section serves as a review of all the material in the book. We'll revisit all the key concepts discussed in order to check how they are affected by the change to complex numbers.

In Chapter 8 we'll discuss the applications of linear algebra. If you've done your job learning the material in the first seven chapters, you'll get to learn all the cool things you can do with linear algebra. Chapter 9 will introduce the basic concepts of probability theory. Chapter 10 contains an introduction to quantum mechanics.

The sections in the book are self-contained so you could read them in any order. Feel free to skip ahead to the parts that you want to learn first. That being said, the material is ordered to provide an optimal knowing-what-you-need-to-know-before-learning-what-you-want-to-know experience. If you're new to linear algebra, it would be best to read them in order. If you find yourself stuck on a concept at some point, refer to the concept maps to see if you're missing some prerequisites and flip to the section of the book that will help you fill in the knowledge gaps accordingly.

www.ebook3000.com

Difficulty level

In terms of difficulty of content, I must prepare you to get ready for some serious uphill pushes. As your personal "trail guide" up the "mountain" of linear algebra, it's my obligation to warn you about the difficulties that lie ahead, so you can mentally prepare for a good challenge.

Linear algebra is a difficult subject because it requires developing your computational skills, your geometrical intuition, and your abstract thinking. The computational aspects of linear algebra are not particularly difficult, but they can be boring and repetitive. You'll have to carry out hundreds of steps of basic arithmetic. The geometrical problems you'll be exposed to in Chapter 5 can be difficult at first, but will get easier once you learn to draw diagrams and develop your geometric reasoning. The theoretical aspects of linear algebra are difficult because they require a new way of thinking, which resembles what doing "real math" is like. You must not only understand and use the material, but also know how to *prove* mathematical statements using the definitions and properties of math objects.

In summary, much toil awaits you as you learn the concepts of linear algebra, but the effort is totally worth it. All the brain sweat you put into understanding vectors and matrices will lead to mind-expanding insights. You will reap the benefits of your efforts for the rest of your life as your knowledge of linear algebra will open many doors for you.

Chapter 1

Math fundamentals

In this chapter we'll review the fundamental ideas of mathematics which are the prerequisites for learning linear algebra. We'll define the different types of numbers and the concept of a function, which is a transformation that takes numbers as inputs and produces numbers as outputs. Linear algebra is the extension of these ideas to many dimensions: instead of "doing math" with numbers and functions, in linear algebra we'll be "doing math" with vectors and linear transformations.

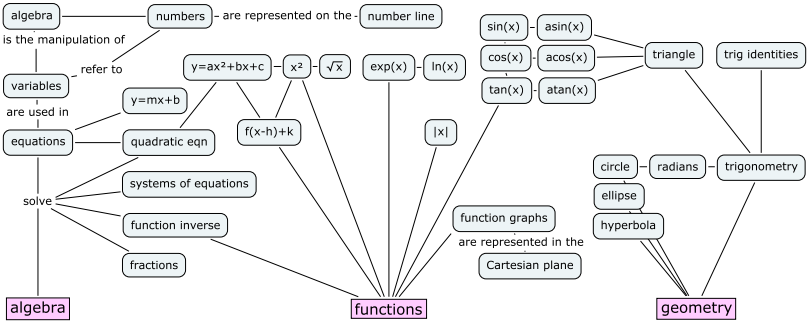


Figure 1.1: A concept map showing the mathematical topics covered in this chapter. We'll learn about how to solve equations using algebra, how to model the world using functions, and some important facts about geometry. The material in this chapter is required for your understanding of the more advanced topics in this book.

1.1 Solving equations

Most math skills boil down to being able to manipulate and solve equations. Solving an equation means finding the value of the unknown in the equation.

Check this shit out:

$$x^2 - 4 = 45.$$

To solve the above equation is to answer the question "What is x?" More precisely, we want to find the number that can take the place of x in the equation so that the equality holds. In other words, we're asking,

"Which number times itself minus four gives 45?"

That is quite a mouthful, don't you think? To remedy this verbosity, mathematicians often use specialized mathematical symbols. The problem is that these specialized symbols can be very confusing. Sometimes even the simplest math concepts are inaccessible if you don't know what the symbols mean.

What are your feelings about math, dear reader? Are you afraid of it? Do you have anxiety attacks because you think it will be too difficult for you? Chill! Relax, my brothers and sisters. There's nothing to it. Nobody can magically guess what the solution to an equation is immediately. To find the solution, you must break the problem down into simpler steps.

To find x, we can manipulate the original equation, transforming it into a different equation (as true as the first) that looks like this:

$$x \;=\; \text{only numbers.}$$

That's what it means to *solve*. The equation is solved because you can type the numbers on the right-hand side of the equation into a calculator and obtain the numerical value of x that you're seeking.

By the way, before we continue our discussion, let it be noted: the equality symbol (=) means that all that is to the left of = is equal to all that is to the right of =. To keep this equality statement true, **for every change you apply to the left side of the equation, you must apply the same change to the right side of the equation**.

To find x, we need to correctly manipulate the original equation into its final form, simplifying it in each step. The only requirement is that the manipulations we make transform one true equation into another true equation. Looking at our earlier example, the first simplifying step is to add the number four to both sides of the equation:

$$x^2 - 4 + 4 = 45 + 4,$$

which simplifies to

$$x^2 = 49.$$

The expression looks simpler, yes? How did I know to perform this operation? I was trying to "undo" the effects of the operation -4. We undo an operation by applying its *inverse*. In the case where the operation is subtraction of some amount, the inverse operation is the addition of the same amount. We'll learn more about function inverses in Section 1.4 (page 18).

We're getting closer to our goal, namely to *isolate* x on one side of the equation, leaving only numbers on the other side. The next step is to undo the square x^2 operation. The inverse operation of squaring a number x^2 is to take the square root $\sqrt{\ }$ so this is what we'll do next. We obtain

$$\sqrt{x^2} = \sqrt{49}.$$

Notice how we applied the square root to both sides of the equation? If we don't apply the same operation to both sides, we'll break the equality!

The equation $\sqrt{x^2} = \sqrt{49}$ simplifies to

$$|x| = 7.$$

What's up with the vertical bars around x? The notation $|x|$ stands for the *absolute value* of x, which is the same as x except we ignore the sign. For example $|5| = 5$ and $|-5| = 5$, too. The equation $|x| = 7$ indicates that both $x = 7$ and $x = -7$ satisfy the equation $x^2 = 49$. Seven squared is 49, and so is $(-7)^2 = 49$ because two negatives cancel each other out.

We're done since we isolated x. The final solutions are

$$x = 7 \qquad \text{or} \qquad x = -7.$$

Yes, there are *two* possible answers. You can check that both of the above values of x satisfy the initial equation $x^2 - 4 = 45$.

If you are comfortable with all the notions of high school math and you feel you could have solved the equation $x^2 - 4 = 45$ on your own, then you should consider skipping ahead to Chapter 2. If on the other hand you are wondering how the squiggle killed the power two, then this chapter is for you! In the following sections we will review all the essential concepts from high school math that you will need to power through the rest of this book. First, let me tell you about the different kinds of numbers.

1.2 Numbers

In the beginning, we must define the main players in the world of math: numbers.

Definitions

Numbers are the basic objects we use to calculate things. Mathematicians like to classify the different kinds of number-like objects into *sets*:

- The natural numbers: $\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, 7, \dots\}$
- The integers: $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
- The rational numbers: $\mathbb{Q} = \{\frac{5}{3}, \frac{22}{7}, 1.5, 0.125, -7, \dots\}$
- The real numbers: $\mathbb{R} = \{-1, 0, 1, \sqrt{2}, e, \pi,\ 4.94\dots,\ \dots\}$
- The complex numbers: $\mathbb{C} = \{-1, 0, 1, i, 1 + i, 2 + 3i, \dots\}$

These categories of numbers should be somewhat familiar to you. Think of them as neat classification labels for everything that you would normally call a number. Each item in the above list is a *set*. A set is a collection of items of the same kind. Each collection has a name and a precise definition. Note also that each of the sets in the list *contains* all the sets above it. For now, we don't need to go into the details of sets and set notation (page 73), but we do need to be aware of the different sets of numbers.

Why do we need so many different sets of numbers? The answer is partly historical and partly mathematical. Each set of numbers is associated with more and more advanced mathematical problems.

The simplest numbers are the natural numbers $\mathbb{N}$, which are sufficient for all your math needs if all you are going to do is *count* things. How many goats? Five goats here and six goats there so the total is 11 goats. The sum of any two natural numbers is also a natural number.

As soon as you start using *subtraction* (the inverse operation of addition), you start running into negative numbers, which are numbers outside the set of natural numbers. If the only mathematical operations you will ever use are *addition* and *subtraction*, then the set of integers $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ will be sufficient. Think about it. Any integer plus or minus any other integer is still an integer.

You can do a lot of interesting math with integers. There is an entire field in math called *number theory* that deals with integers. However, to restrict yourself solely to integers is somewhat limiting. You can't use the notion of 2.5 goats for example. The menu at Rotisserie Romados, which offers $\frac{1}{4}$ of a chicken, would be completely confusing.

If you want to use division in your mathematical calculations, you'll need the rationals $\mathbb{Q}$. The rationals are the set of *fractions* of integers:

$$\mathbb{Q} = \left\{ \text{all } z \text{ such that } z = \frac{x}{y} \text{ where } x \text{ and } y \text{ are in } \mathbb{Z}, \text{ and } y \neq 0 \right\}.$$

You can add, subtract, multiply, and divide rational numbers, and the result will always be a rational number. However, even the rationals are not enough for all of math!

In geometry, we can obtain *irrational* quantities like $\sqrt{2}$ (the diagonal of a square with side 1) and π (the ratio between a circle's circumference and its diameter). There are no integers x and y such that $\sqrt{2} = \frac{x}{y}$. Therefore, $\sqrt{2}$ is not part of the set $\mathbb{Q}$, and we say that $\sqrt{2}$ is *irrational*. An irrational number has an infinitely long decimal expansion that doesn't repeat. For example, $\pi = 3.141592653589793\ldots$ where the dots indicate that the decimal expansion of π continues all the way to infinity.

Adding the irrational numbers to the rationals gives us all the useful numbers, which we call the set of real numbers $\mathbb{R}$. The set $\mathbb{R}$ contains the integers, the fractions $\mathbb{Q}$, as well as irrational numbers like $\sqrt{2} = 1.4142135\ldots$. By using the reals you can compute pretty much anything you want. From here on in the text, when I say *number*, I mean an element of the set of real numbers $\mathbb{R}$.

The only thing you can't do with the reals is take the square root of a negative number—you need the complex numbers $\mathbb{C}$ for that. We defer the discussion on $\mathbb{C}$ until the end of Chapter 3.

Operations on numbers

Addition

You can add and subtract numbers. I will assume you are familiar with this kind of stuff:

$$2 + 5 = 7, \quad 45 + 56 = 101, \quad 65 - 66 = -1, \quad 9\,999 + 1 = 10\,000.$$

It can help visual learners to picture numbers as lengths measured out on the *number line*. Adding numbers is like adding sticks together: the resulting stick has a length equal to the sum of the lengths of the constituent sticks.

Addition is *commutative*, which means that $a + b = b + a$. It is also *associative*, which means that if you have a long summation like $a + b + c$ you can compute it in any order $(a + b) + c$ or $a + (b + c)$ and you'll get the same answer.

Subtraction is the inverse operation of addition.

Multiplication

You can also multiply numbers together.

$$ab = \underbrace{a + a + \cdots + a}_{b \text{ times}} = \underbrace{b + b + \cdots + b}_{a \text{ times}}.$$

Note that multiplication can be defined in terms of repeated addition.

The visual way to think about multiplication is as an area calculation. The area of a rectangle of base a and height b is equal to ab. A rectangle with a height equal to its base is a square, and this is why we call $aa = a^2$ "a squared."

Multiplication of numbers is also commutative, $ab = ba$; and associative, $abc = (ab)c = a(bc)$. In modern notation, no special symbol is used to denote multiplication; we simply put the two factors next to each other and say the multiplication is *implicit*. Some other ways to denote multiplication are $a \cdot b$, $a \times b$, and, on computer systems, $a * b$.

Division

Division is the inverse operation of multiplication.

$$a/b = \frac{a}{b} = \text{ one } b^{\text{th}} \text{ of } a.$$

Whatever a is, you need to divide it into b equal parts and take one such part. Some texts denote division as $a \div b$.

Note that you cannot divide by 0. Try it on your calculator or computer. It will say "`error divide by zero`" because this action simply doesn't make sense. After all, what would it mean to divide something into zero equal parts?

Exponentiation

Often an equation calls for us to multiply things together many times. The act of multiplying a number by itself many times is called *exponentiation*, and we denote this operation as a superscript:

$$a^b = \underbrace{aaa\cdots a}_{b \text{ times}}.$$

We can also encounter negative exponents. The negative in the exponent does not mean "subtract," but rather "divide by":

$$a^{-b} = \frac{1}{a^b} = \frac{1}{\underbrace{aaa\cdots a}_{b \text{ times}}}.$$

Fractional exponents describe square-root-like operations:

$$a^{\frac{1}{2}} \equiv \sqrt{a} \equiv \sqrt[2]{a}, \quad a^{\frac{1}{3}} \equiv \sqrt[3]{a}, \quad a^{\frac{1}{4}} \equiv \sqrt[4]{a} = a^{\frac{1}{2}\frac{1}{2}} = \left(a^{\frac{1}{2}}\right)^{\frac{1}{2}} = \sqrt{\sqrt{a}}.$$

Square root $\sqrt{x}$ is the inverse operation of x^2. Similarly, for any n we define the function $\sqrt[n]{x}$ (the n^{th} root of x) to be the inverse function of x^n.

It's worth clarifying what "taking the n^{th} root" means and understanding when to use this operation. The n^{th} root of a is a number which, when multiplied together n times, will give a. For example, a cube root satisfies

$$\sqrt[3]{a}\sqrt[3]{a}\sqrt[3]{a} = \left(\sqrt[3]{a}\right)^3 = a = \sqrt[3]{a^3}.$$

Do you see why $\sqrt[3]{x}$ and x^3 are inverse operations?

The fractional exponent notation makes the meaning of roots much more explicit. The n^{th} root of a can be denoted in two equivalent ways:

$$\sqrt[n]{a} \equiv a^{\frac{1}{n}}.$$

The symbol "$\equiv$" stands for "is equivalent to" and is used when two mathematical objects are identical. Equivalence is a stronger relation than equality. Writing $\sqrt[n]{a} = a^{\frac{1}{n}}$ indicates we've found two mathematical expressions (the left-hand side and the right-hand side of the equality) that happen to be equal to each other. It is more mathematically precise to write $\sqrt[n]{a} \equiv a^{\frac{1}{n}}$, which tells us $\sqrt[n]{a}$ and $a^{\frac{1}{n}}$ are two different ways of denoting the *same* mathematical object.

The n^{th} root of a is equal to one n^{th} of a with respect to multiplication. To find the whole number, multiply the number $a^{\frac{1}{n}}$ times itself n times:

$$\underbrace{a^{\frac{1}{n}}a^{\frac{1}{n}}a^{\frac{1}{n}}a^{\frac{1}{n}}\cdots a^{\frac{1}{n}}a^{\frac{1}{n}}}_{n \text{ times}} = \left(a^{\frac{1}{n}}\right)^n = a^{\frac{n}{n}} = a^1 = a.$$

The n-fold product of $\frac{1}{n}$-fractional exponents of any number produces that number with exponent one, therefore the inverse operation of $\sqrt[n]{x}$ is x^n.

The commutative law of multiplication $ab = ba$ implies that we can see any fraction $\frac{a}{b}$ in two different ways: $\frac{a}{b} = a\frac{1}{b} = \frac{1}{b}a$. We multiply by a then divide the result by b, or first we divide by b and then multiply the result by a. Similarly, when we have a fraction in the exponent, we can write the answer in two equivalent ways:

$$a^{\frac{2}{3}} = \sqrt[3]{a^2} = (\sqrt[3]{a})^2, \quad a^{-\frac{1}{2}} = \frac{1}{a^{\frac{1}{2}}} = \frac{1}{\sqrt{a}}, \quad a^{\frac{m}{n}} = \left(\sqrt[n]{a}\right)^m = \sqrt[n]{a^m}.$$

Make sure the above notation makes sense to you. As an exercise, try computing $5^{\frac{4}{3}}$ on your calculator and check that you obtain $8.54987973\ldots$ as the answer.

Operator precedence

There is a standard convention for the order in which mathematical operations must be performed. The basic algebra operations have the following precedence:

1. Exponents and roots
2. Products and divisions
3. Additions and subtractions

For instance, the expression $5 \times 3^2 + 13$ is interpreted as "first find the square of 3, then multiply it by 5, and then add 13." Parenthesis are needed to carry out the operations in a different order: to multiply 5 times 3 first and *then* take the square, the equation should read $(5 \times 3)^2 + 13$, where parenthesis indicate that the square acts on (5×3) as a whole and not on 3 alone.

Other operations

We can define all kinds of operations on numbers. The above three are special operations since they feel simple and intuitive to apply, but we can also define arbitrary transformations on numbers. We call these transformations *functions*. Before we learn about functions, let's first cover variables.

1.3 Variables

In math we use a lot of *variables*, which are placeholder names for *any* number or unknown.

Example Your friend invites you to a party and offers you to drink from a weirdly shaped shooter glass. You can't quite tell if it holds 25 ml of vodka or 50 ml or some amount in between. Since it's a mystery how much booze each shot contains, you shrug your shoulders and say there's x ml in there. The night happens. So how much did you drink? If you had three shots, then you drank $3x$ ml of vodka. If you want to take it a step further, you can say you drank n shots, making the total amount of alcohol you consumed nx ml.

Variables allow us to talk about quantities without knowing the details. This is *abstraction* and it is very powerful stuff: it allows you to get drunk without knowing how drunk exactly!

Variable names

There are common naming patterns for variables:

- x: general name for the unknown in equations (also used to denote a function's input, as well as an object's position in physics problems)
- v: velocity in physics problems
- θ, φ: the Greek letters *theta* and *phi* are used to denote angles
- x_i, x_f: denote an object's initial and final positions in physics problems
- X: a random variable in probability theory
- C: costs in business along with P for profit, and R for revenue

Variable substitution

We can often *change variables* and replace one unknown variable with another to simplify an equation. For example, say you don't feel comfortable around square roots. Every time you see a square root, you freak out until one day you find yourself taking an exam trying to solve for x in the following equation:

$$\frac{6}{5-\sqrt{x}} = \sqrt{x}.$$

Don't freak out! In crucial moments like this, substitution can help with your root phobia. Just write, "Let $u = \sqrt{x}$" on your exam, and voila, you can rewrite the equation in terms of the variable u:

$$\frac{6}{5-u} = u,$$

which contains no square roots.

The next step to solve for u is to undo the division operation. Multiply both sides of the equation by $(5-u)$ to obtain

$$\frac{6}{5-u}(5-u) = u(5-u),$$

which simplifies to

$$6 = 5u - u^2.$$

This can be rewritten as a quadratic equation, $u^2 - 5u + 6 = 0$. Next, we can *factor* the quadratic to obtain the equation $(u-2)(u-3) = 0$, for which $u_1 = 2$ and $u_2 = 3$ are the solutions. The last step is to convert our u-answers into x answers by using $u = \sqrt{x}$, which is equivalent to $x = u^2$. The final answers are $x_1 = 2^2 = 4$ and $x_2 = 3^2 = 9$. Try plugging these x values into the original square root equation to verify that they satisfy it.

Compact notation

Symbolic manipulation is a powerful tool because it allows us to manage complexity. Say you're solving a physics problem in which you're told the mass of an object is $m = 140$ kg. If there are many steps in the calculation, would you rather use the number 140 kg in each step, or the shorter variable m? It's much easier in the long run to use the variable m throughout your calculation, and wait until the last step to substitute the value 140 kg when computing the final answer.

1.4 Functions and their inverses

As we saw in the section on solving equations, the ability to "undo" functions is a key skill for solving equations.

Example Suppose we're solving for x in the equation

$$f(x) = c,$$

where f is some function and c is some constant. Our goal is to isolate x on one side of the equation, but the function f stands in our way.

By using the inverse function (denoted f^{-1}) we "undo" the effects of f. Then we apply the inverse function f^{-1} to both sides of the equation to obtain

$$f^{-1}(f(x)) = x = f^{-1}(c).$$

By definition, the inverse function f^{-1} performs the opposite action of the function f so together the two functions cancel each other out. We have $f^{-1}(f(x)) = x$ for any number x.

Provided everything is kosher (the function f^{-1} must be defined for the input c), the manipulation we made above is valid and we have obtained the answer $x = f^{-1}(c)$.

The above example introduces the notation f^{-1} for denoting the function's *inverse*. This notation is borrowed from the notion of inverse numbers: multiplication by the number a^{-1} is the inverse operation of multiplication by the number a: $a^{-1}ax = 1x = x$. In the case of functions, however, the negative-one exponent does not refer to "one over-$f(x)$" as in $\frac{1}{f(x)} = (f(x))^{-1}$; rather, it refers to the function's inverse. In other words, the number $f^{-1}(y)$ is equal to the number x such that $f(x) = y$.

Be careful: sometimes applying the inverse leads to multiple solutions. For example, the function $f(x) = x^2$ maps two input values (x and $-x$) to the same output value $x^2 = f(x) = f(-x)$. The inverse function of $f(x) = x^2$ is $f^{-1}(x) = \sqrt{x}$, but both $x = +\sqrt{c}$

and $x = -\sqrt{c}$ are solutions to the equation $x^2 = c$. In this case, this equation's solutions can be indicated in shorthand notation as $x = \pm\sqrt{c}$.

Formulas

Here is a list of common functions and their inverses:

$$\begin{array}{rcl} \text{function } f(x) & \Leftrightarrow & \text{inverse } f^{-1}(x) \\ x+2 & \Leftrightarrow & x-2 \\ 2x & \Leftrightarrow & \frac{1}{2}x \\ -x & \Leftrightarrow & -x \\ x^2 & \Leftrightarrow & \pm\sqrt{x} \\ 2^x & \Leftrightarrow & \log_2(x) \\ 3x+5 & \Leftrightarrow & \frac{1}{3}(x-5) \\ a^x & \Leftrightarrow & \log_a(x) \\ \exp(x) \equiv e^x & \Leftrightarrow & \ln(x) \equiv \log_e(x) \\ \sin(x) & \Leftrightarrow & \sin^{-1}(x) \equiv \arcsin(x) \\ \cos(x) & \Leftrightarrow & \cos^{-1}(x) \equiv \arccos(x) \end{array}$$

The function-inverse relationship is *reflexive*—if you see a function on one side of the above table (pick a side, any side), you'll find its inverse on the opposite side.

Example

Let's say your teacher doesn't like you and right away, on the first day of class, he gives you a serious equation and tells you to find x:

$$\log_5\left(3+\sqrt{6\sqrt{x}-7}\right) = 34 + \sin(5.5) - \Psi(1).$$

See what I mean when I say the teacher doesn't like you?

First, note that it doesn't matter what Ψ (the capital Greek letter *psi*) is, since x is on the other side of the equation. You can keep copying $\Psi(1)$ from line to line, until the end, when you throw the ball back to the teacher. "My answer is in terms of *your* variables, dude. *You* go figure out what the hell Ψ is since you brought it up in the first place!" By the way, it's not actually recommended to quote me verbatim should a situation like this arise. The same goes with $\sin(5.5)$. If you don't have a calculator handy, don't worry about it. Keep the expression $\sin(5.5)$ instead of trying to find its numerical

value. In general, try to work with variables as much as possible and leave the numerical computations for the last step.

Okay, enough beating about the bush. Let's just find x and get it over with! On the right-hand side of the equation, we have the sum of a bunch of terms with no x in them, so we'll leave them as they are. On the left-hand side, the outermost function is a logarithm base 5. Cool. Looking at the table of inverse functions we find the exponential function is the inverse of the logarithm: $a^x \Leftrightarrow \log_a(x)$. To get rid of $\log_5$, we must apply the exponential function base 5 to both sides:

$$5^{\log_5\left(3+\sqrt{6\sqrt{x}-7}\right)} = 5^{34+\sin(5.5)-\Psi(1)},$$

which simplifies to

$$3+\sqrt{6\sqrt{x}-7} = 5^{34+\sin(5.5)-\Psi(1)},$$

since 5^x cancels $\log_5 x$.

From here on, it is going to be as if Bruce Lee walked into a place with lots of bad guys. Addition of 3 is undone by subtracting 3 on both sides:

$$\sqrt{6\sqrt{x}-7} = 5^{34+\sin(5.5)-\Psi(1)} - 3.$$

To undo a square root we take the square:

$$6\sqrt{x}-7 = \left(5^{34+\sin(5.5)-\Psi(1)} - 3\right)^2.$$

Add 7 to both sides,

$$6\sqrt{x} = \left(5^{34+\sin(5.5)-\Psi(1)} - 3\right)^2 + 7,$$

divide by 6

$$\sqrt{x} = \frac{1}{6}\left(\left(5^{34+\sin(5.5)-\Psi(1)} - 3\right)^2 + 7\right),$$

and square again to find the final answer:

$$x = \left[\frac{1}{6}\left(\left(5^{34+\sin(5.5)-\Psi(1)} - 3\right)^2 + 7\right)\right]^2.$$

Did you see what I was doing in each step? Next time a function stands in your way, hit it with its inverse so it knows not to challenge you ever again.

Discussion

The recipe I have outlined above is not universally applicable. Sometimes x isn't alone on one side. Sometimes x appears in several places in the same equation. In these cases, you can't effortlessly work your way, Bruce Lee-style, clearing bad guys and digging toward x—you need other techniques.

The bad news is there's no general formula for solving complicated equations. The good news is the above technique of "digging toward x" is sufficient for 80% of what you are going to be doing. You can get another 15% if you learn how to solve the quadratic equation (page 25):

$$ax^2 + bx + c = 0.$$

Solving third-degree polynomial equations like $ax^3 + bx^2 + cx + d = 0$ with pen and paper is also possible, but at this point you might as well start using a computer to solve for the unknowns.

There are all kinds of other equations you can learn how to solve: equations with multiple variables, equations with logarithms, equations with exponentials, and equations with trigonometric functions. The principle of "digging" toward the unknown by applying inverse functions is the key for solving all these types of equations, so be sure to practice using it.

1.5 Basic rules of algebra

It's important that you know the general rules for manipulating numbers and variables, a process otherwise known as—you guessed it—*algebra*. This little refresher will cover these concepts to make sure you're comfortable on the algebra front. We'll also review some important algebraic tricks, like *factoring* and *completing the square*, which are useful when solving equations.

When an expression contains multiple things added together, we call those things *terms*. Furthermore, terms are usually composed of many things multiplied together. When a number x is obtained as the product of other numbers like $x = abc$, we say "x factors into a, b, and c." We call a, b, and c the *factors* of x.

Given any four numbers a, b, c, and d, we can apply the following algebraic properties:

1. Associative property: $a + b + c = (a + b) + c = a + (b + c)$ and $abc = (ab)c = a(bc)$
2. Commutative property: $a + b = b + a$ and $ab = ba$
3. Distributive property: $a(b + c) = ab + ac$

We use the distributive property every time we *expand* brackets. For example $a(b + c + d) = ab + ac + ad$. The brackets, also known as parentheses, indicate the expression $(b + c + d)$ must be treated as a whole: a factor that consists of three terms. Multiplying this expression by a is the same as multiplying each term by a.

The opposite operation of expanding is called *factoring*, which consists of rewriting the expression with the common parts taken out in front of a bracket: $ab + ac = a(b + c)$. In this section, we'll discuss both of these operations and illustrate what they're capable of.

Expanding brackets

The distributive property is useful when dealing with polynomials:

$$(x + 3)(x + 2) = x(x + 2) + 3(x + 2) = x^2 + x2 + 3x + 6.$$

We can use the commutative property on the second term $x2 = 2x$, then combine the two x terms into a single term to obtain

$$(x + 3)(x + 2) = x^2 + 5x + 6.$$

Let's look at this operation in its abstract form:

$$(x + a)(x + b) = x^2 + (a + b)x + ab.$$

The product of two linear terms (expressions of the form $x + ?$) is equal to a quadratic expression. Observe that the middle term on the right-hand side contains the *sum* of the two constants on the left-hand side $(a + b)$, while the third term contains their product ab.

It is very common for people to confuse these terms. If you are ever confused about an algebraic expression, go back to the distributive property and expand the expression using a step-by-step approach. As a second example, consider this slightly-more-complicated algebraic expression and its expansion:

$$\begin{aligned}(x + a)(bx^2 + cx + d) &= x(bx^2 + cx + d) + a(bx^2 + cx + d)\\ &= bx^3 + cx^2 + dx + abx^2 + acx + ad\\ &= bx^3 + (c + ab)x^2 + (d + ac)x + ad.\end{aligned}$$

Note how all terms containing x^2 are grouped into a one term, and all terms containing x are grouped into another term. We use this pattern when dealing with expressions containing different powers of x.

Example Suppose we are asked to solve for t in the equation

$$7(3 + 4t) = 11(6t - 4).$$

Since the unknown t appears on both sides of the equation, it is not immediately obvious how to proceed.

To solve for t, we must bring all t terms to one side and all constant terms to the other side. First, expand the two brackets to obtain

$$21 + 28t = 66t - 44.$$

Then move things around to relocate all ts to the equation's right-hand side and all constants to the left-hand side:

$$21 + 44 = 66t - 28t.$$

We see t is contained in both terms on the right-hand side, so we can rewrite the equation as

$$21 + 44 = (66 - 28)t.$$

The answer is within close reach: $t = \frac{21+44}{66-28} = \frac{65}{38}$.

Factoring

Factoring involves taking out the common part(s) of a complicated expression in order to make the expression more compact. Suppose you're given the expression $6x^2y + 15x$ and must simplify it by taking out common factors. The expression has two terms and each term can be split into its constituent factors to obtain

$$6x^2y + 15x = (3)(2)(x)(x)y + (5)(3)x.$$

Since factors x and 3 appear in both terms, we can *factor them out* to the front like this:

$$6x^2y + 15x = 3x(2xy + 5).$$

The expression on the right shows $3x$ is common to both terms.

Here's another example where factoring is used:

$$2x^2y + 2x + 4x = 2x(xy + 1 + 2) = 2x(xy + 3).$$

Quadratic factoring

When dealing with a quadratic function, it is often useful to rewrite the function as a product of two factors. Suppose you're given the quadratic function $f(x) = x^2 - 5x + 6$ and asked to describe its properties. What are the *roots* of this function? In other words, for what values of x is this function equal to zero? For which values of x is the function positive, and for which x values is the function negative?

Factoring the expression $x^2 - 5x + 6$ will help us see the properties of the function more clearly. To *factor* a quadratic expression is to express it as the product of two factors:

$$f(x) = x^2 - 5x + 6 = (x - 2)(x - 3).$$

We now see at a glance the solutions (roots) are $x_1 = 2$ and $x_2 = 3$. We can also see for which x values the function will be overall positive: for $x > 3$, both factors will be positive, and for $x < 2$ both factors will be negative, and a negative times a negative gives a positive. For values of x such that $2 < x < 3$, the first factor will be positive, and the second factor negative, making the overall function negative.

For certain simple quadratics like the one above, you can simply *guess* what the factors will be. For more complicated quadratic expressions, you'll need to use the quadratic formula (page 25), which will be the subject of the next section. For now let us continue with more algebra tricks.

Completing the square

Any quadratic expression $Ax^2 + Bx + C$ can be rewritten in the form $A(x - h)^2 + k$ for some constants h and k. This process is called *completing the square* due to the reasoning we follow to find the value of k. The constants h and k can be interpreted geometrically as the horizontal and vertical shifts in the graph of the basic quadratic function. The graph of the function $f(x) = A(x - h)^2 + k$ is the same as the graph of the function $f(x) = Ax^2$ except it is shifted h units to the right and k units upward. We will discuss the geometrical meaning of h and k in more detail in Section 1.9 (page 49). For now, let's focus on the algebra steps.

Let's try to find the values of k and h needed to complete the square in the expression $x^2 + 5x + 6$. We start from the assumption that the two expressions are equal, and then expand the bracket to obtain

$$\underline{x^2} + 5x + 6 = A(x-h)^2 + k = A(x^2 - 2hx + h^2) + k = \underline{Ax^2} - 2Ahx + Ah^2 + k.$$

Observe the structure in the above equation. On both sides of the equality there is one term which contains x^2 (the quadratic term), one term that contains x^1 (the linear term), and some constant terms. By focusing on the quadratic terms on both sides of the equation (they are underlined) we see $A = 1$, so we can rewrite the equation as

$$x^2 + \underline{5x} + 6 = x^2 \underline{- 2hx} + h^2 + k.$$

Next we look at the linear terms (underlined) and infer $h = -2.5$. After rewriting, we obtain an equation in which k is the only unknown:

$$x^2 + 5x + \underline{6} = x^2 - 2(-2.5)x + \underline{(-2.5)^2 + k}.$$

We must pick a value of k that makes the constant terms equal:

$$k = 6 - (-2.5)^2 = 6 - (2.5)^2 = 6 - \left(\frac{5}{2}\right)^2 = 6 \times \frac{4}{4} - \frac{25}{4} = \frac{24 - 25}{4} = \frac{-1}{4}.$$

After completing the square we obtain

$$x^2 + 5x + 6 = (x + 2.5)^2 - \frac{1}{4}.$$

The right-hand side of the expression above tells us our function is equivalent to the basic function x^2, shifted 2.5 units to the left and $\frac{1}{4}$ units down. This would be very useful information if you ever had to draw the graph of this function—you could simply plot the basic graph of x^2 and then shift it appropriately.

It is important you become comfortable with this procedure for completing the square. It is not extra difficult, but it does require you to think carefully about the unknowns h and k and to choose their values appropriately. There is no general formula for finding k, but you can remember the following simple shortcut for finding h. Given an equation $Ax^2 + Bx + C = A(x - h)^2 + k$, we have $h = \frac{-B}{2A}$. Using this shortcut will save you some time, but you will still have to go through the algebra steps to find k.

Take out a pen and a piece of paper now (yes, right now!) and verify that you can correctly complete the square in these expressions: $x^2 - 6x + 13 = (x - 3)^2 + 4$ and $x^2 + 4x + 1 = (x + 2)^2 - 3$.

1.6 Solving quadratic equations

What would you do if asked to solve for x in the quadratic equation $x^2 = 45x + 23$? This is called a *quadratic equation* since it contains the unknown variable x squared. The name comes from the Latin *quadratus*, which means square. Quadratic equations appear often, so mathematicians created a general formula for solving them. In this section, we'll learn about this formula and use it to put some quadratic equations in their place.

Before we can apply the formula, we need to rewrite the equation we are trying to solve in the following form:

$$ax^2 + bx + c = 0.$$

We reach this form—called the *standard form* of the quadratic equation—by moving all the numbers and xs to one side and leaving only 0 on the other side. For example, to transform the quadratic equation $x^2 = 45x + 23$ into standard form, subtract $45x + 23$ from both sides of the equation to obtain $x^2 - 45x - 23 = 0$. What are the values of x that satisfy this formula?

Claim

The solutions to the equation $ax^2 + bx + c = 0$ are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad \text{and} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Let's see how these formulas are used to solve $x^2 - 45x - 23 = 0$. Finding the two solutions requires the simple mechanical task of identifying $a = 1$, $b = -45$, and $c = -23$ and plugging these values into the formulas:

$$x_1 = \frac{45 + \sqrt{45^2 - 4(1)(-23)}}{2} = 45.5054\ldots,$$

$$x_2 = \frac{45 - \sqrt{45^2 - 4(1)(-23)}}{2} = -0.5054\ldots.$$

Verify using your calculator that both of the values above satisfy the original equation $x^2 = 45x + 23$.

Proof of claim

This is an important proof. I want you to see how we can *derive* the quadratic formula from first principles because this knowledge will help you understand the formula. The proof will use the completing-the-square technique from the previous section.

Starting with $ax^2 + bx + c = 0$, first move c to the other side of the equation:

$$ax^2 + bx = -c.$$

Divide by a on both sides:

$$x^2 + \frac{b}{a}x = -\frac{c}{a}.$$

Now *complete the square* on the left-hand side by asking, "What are the values of h and k that satisfy the equation

$$(x - h)^2 + k = x^2 + \frac{b}{a}x \quad ?"$$

To find the values for h and k, we'll expand the left-hand side to obtain $(x - h)^2 + k = x^2 - 2hx + h^2 + k$. We can now identify h by looking at the coefficients in front of x on both sides of the equation. We have $-2h = \frac{b}{a}$ and hence $h = -\frac{b}{2a}$.

Let's see what we have so far:

$$\left(x+\frac{b}{2a}\right)^2=\left(x+\frac{b}{2a}\right)\left(x+\frac{b}{2a}\right)=x^2+\frac{b}{2a}x+x\frac{b}{2a}+\frac{b^2}{4a^2}=x^2+\frac{b}{a}x+\frac{b^2}{4a^2}.$$

To determine k, we need to move that last term to the other side:

$$\left(x+\frac{b}{2a}\right)^2-\frac{b^2}{4a^2}=x^2+\frac{b}{a}x.$$

We can continue with the proof where we left off:

$$x^2+\frac{b}{a}x=-\frac{c}{a}.$$

Replace the left-hand side with the complete-the-square expression and obtain

$$\left(x+\frac{b}{2a}\right)^2-\frac{b^2}{4a^2}=-\frac{c}{a}.$$

From here on, we can use the standard procedure for solving equations (page 10). Arrange all constants on the right-hand side:

$$\left(x+\frac{b}{2a}\right)^2=-\frac{c}{a}+\frac{b^2}{4a^2}.$$

Next, take the square root of both sides. Since the square function maps both positive and negative numbers to the same value, this step yields two solutions:

$$x+\frac{b}{2a}=\pm\sqrt{-\frac{c}{a}+\frac{b^2}{4a^2}}.$$

Let's take a moment to tidy up the mess under the square root:

$$\sqrt{-\frac{c}{a}+\frac{b^2}{4a^2}}=\sqrt{-\frac{(4a)c}{(4a)a}+\frac{b^2}{4a^2}}=\sqrt{\frac{-4ac+b^2}{4a^2}}=\frac{\sqrt{b^2-4ac}}{2a}.$$

We obtain

$$x+\frac{b}{2a}=\pm\frac{\sqrt{b^2-4ac}}{2a},$$

which is just one step from the final answer,

$$x=\frac{-b}{2a}\pm\frac{\sqrt{b^2-4ac}}{2a}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}.$$

This completes the proof. □

Alternative proof of claim

To have a proof we don't necessarily need to show the derivation of the formula as outlined above. The claim states that x_1 and x_2 are solutions. To prove the claim we can simply plug x_1 and x_2 into the quadratic equation and verify the answers are zero. **Verify this on your own.**

Applications

The Golden Ratio

The golden ratio is an essential proportion in geometry, art, aesthetics, biology, and mysticism, and is usually denoted as $\varphi = \frac{1+\sqrt{5}}{2} = 1.6180339\ldots$. This ratio is determined as the positive solution to the quadratic equation

$$x^2 - x - 1 = 0.$$

Applying the quadratic formula to this equation yields two solutions,

$$x_1 = \frac{1+\sqrt{5}}{2} = \varphi \qquad \text{and} \qquad x_2 = \frac{1-\sqrt{5}}{2} = -\frac{1}{\varphi}\,.$$

You can learn more about the various contexts in which the golden ratio appears from the Wikipedia article on the subject.

Explanations

Multiple solutions

Often, we are interested in only one of the two solutions to the quadratic equation. It will usually be obvious from the context of the problem which of the two solutions should be kept and which should be discarded. For example, the *time of flight* of a ball thrown in the air from a height of 3 metres with an initial velocity of 12 metres per second is obtained by solving the equation $(-4.9)t^2 + 12t + 3 = 0$. The two solutions of the quadratic equation are $t_1 = -0.229$ and $t_2 = 2.678$. The first answer t_1 corresponds to a time in the past so we reject it as invalid. The correct answer is t_2. The ball will hit the ground after $t = 2.678$ seconds.

Relation to factoring

In the previous section we discussed the *quadratic factoring* operation by which we could rewrite a quadratic function as the product of two terms $f(x) = ax^2 + bx + c = (x - x_1)(x - x_2)$. The two numbers x_1

and x_2 are called the *roots* of the function: these points are where the function $f(x)$ touches the x-axis.

You now have the ability to factor any quadratic equation. Use the quadratic formula to find the two solutions, x_1 and x_2, then rewrite the expression as $(x - x_1)(x - x_2)$.

Some quadratic expressions cannot be factored, however. These "unfactorable" expressions correspond to quadratic functions whose graphs do not touch the x-axis. They have no solutions (no roots). There is a quick test you can use to check if a quadratic function $f(x) = ax^2 + bx + c$ has roots (touches or crosses the x-axis) or doesn't have roots (never touches the x-axis). If $b^2 - 4ac > 0$ then the function f has two roots. If $b^2 - 4ac = 0$, the function has only one root, indicating the special case when the function touches the x-axis at only one point. If $b^2 - 4ac < 0$, the function has no roots. In this case the quadratic formula fails because it requires taking the square root of a negative number, which is not allowed. Think about it—how could you square a number and obtain a negative number?

1.7 The Cartesian plane

Named after famous philosopher and mathematician René Descartes, the Cartesian plane is a graphical representation for *pairs* of numbers.

Generally, we call the plane's horizontal axis "the x-axis" and its vertical axis "the y-axis." We put notches at regular intervals on each axis so we can measure distances. Figure 1.2 is an example of an empty Cartesian coordinate system. Think of the coordinate system as an empty canvas. What can you draw on this canvas?

Vectors and points

A *point* $P = (P_x, P_y)$ in the Cartesian plane has an x-coordinate and a y-coordinate. To find this point, start from the origin—the point (0,0)—and move a distance P_x on the x-axis, then move a distance P_y on the y-axis.

Similar to a point, a vector $\vec{v} = (v_x, v_y)$ is a pair of coordinates. Unlike points, we don't necessarily start from the plane's origin when mapping vectors. We draw vectors as arrows that explicitly mark where the vector starts and where it ends. Note that vectors $\vec{v}_2$ and $\vec{v}_3$ illustrated in Figure 1.3 are actually the *same* vector—the "displace left by 1 and down by 2" vector. It doesn't matter where you draw this vector, it will always be the same whether it begins at the plane's origin or elsewhere.

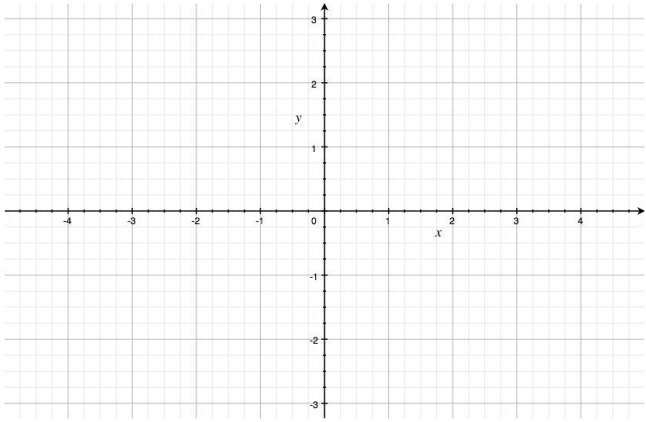


Figure 1.2: The (x, y)-coordinate system, which is also known as the Cartesian plane. Points $P = (P_x, P_y)$, vectors $\vec{v} = (v_x, v_y)$, and graphs of functions $(x, f(x))$ live here.

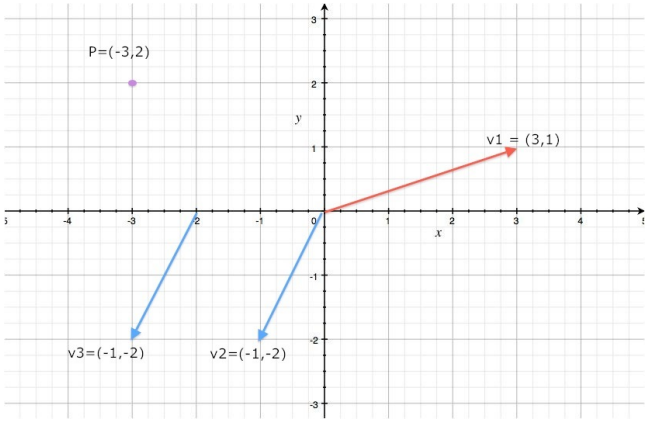


Figure 1.3: A Cartesian plane which shows the point $P = (-3, 2)$ and the vectors $\vec{v}_1 = (3, 1)$ and $\vec{v}_2 = \vec{v}_3 = (-1, -2)$.

Graphs of functions

The Cartesian plane is great for visualizing functions,

$$f : \mathbb{R} \to \mathbb{R}.$$

You can think of a function as a set of input-output pairs $(x, f(x))$. You can *graph* a function by letting the y-coordinate represent the function's output value:

$$(x, y) = (x, f(x)).$$

For example, with the function $f(x) = x^2$, we can pass a line through the set of points

$$(x, y) = (x, x^2),$$

and obtain the graph shown in Figure 1.4.

When plotting functions by setting $y = f(x)$, we use a special terminology for the two axes. The x-axis represents the *independent* variable (the one that varies freely), and the y-axis represents the *dependent* variable $f(x)$, since $f(x)$ depends on x.

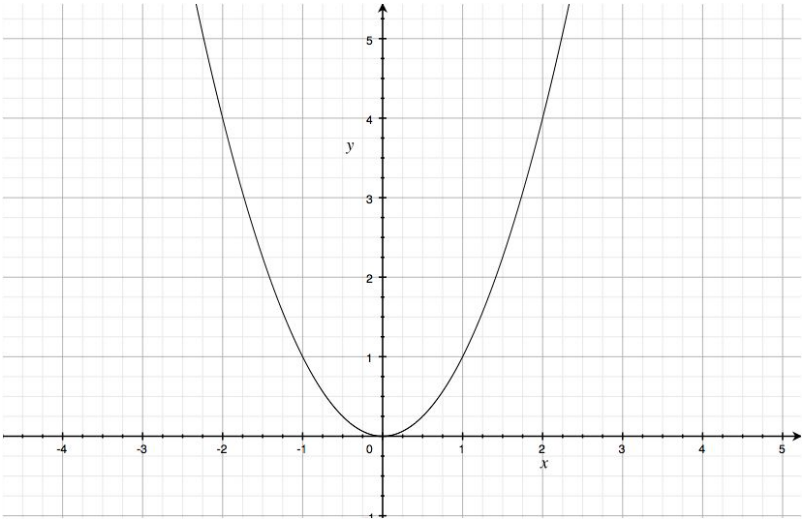


Figure 1.4: The graph of the function $f(x) = x^2$ consists of all pairs of points (x, y) in the Cartesian plane that satisfy $y = x^2$.

To draw the graph of any function $f(x)$, use the following procedure. Imagine making a sweep over all of the possible input values for the function. For each input x, put a point at the coordinates $(x, y) = (x, f(x))$ in the Cartesian plane. Using the graph of a function, you can literally *see* what the function does: the "height" y of the graph at a given x-coordinate tells you the value of the function $f(x)$.

Discussion

To build mathematical intuition, it is essential you understand the graphs of functions. Trying to memorize the definitions and the properties of functions is a difficult task. Remembering what the function "looks like" is comparatively easier. You should spend some time fa-

miliarizing yourself with the graphs of the functions presented in the next section.

1.8 Functions

We need to have a relationship talk. We need to talk about functions. We use functions to describe the relationships between variables. In particular, functions describe how one variable *depends* on another.

For example, the revenue R from a music concert depends on the number of tickets sold n. If each ticket costs \$25, the revenue from the concert can be written *as a function of* n as follows: $R(n) = 25n$. Solving for n in the equation $R(n) = 7000$ tells us the number of ticket sales needed to generate \$7000 in revenue. This is a simple model of a function; as your knowledge of functions builds, you'll learn how to build more detailed models of reality. For instance, if you need to include a 5% processing charge for issuing the tickets, you can update the revenue model to $R(n) = 0.95 \cdot 25 \cdot n$. If the estimated cost of hosting the concert is $C = \$2000$, then the profit from the concert P can be modelled as

$$\begin{aligned} P(n) &= R(n) \ - \ C \\ &= 0.95 \cdot \$25 \cdot n \ - \ \$2000 \end{aligned}$$

The function $P(n) = 23.75n - 2000$ models the profit from the concert as a function of the number of tickets sold. This is a pretty good model already, and you can always update it later on as you find out more information.

The more functions you know, the more tools you have for modelling reality. To "know" a function, you must be able to understand and connect several of its aspects. First you need to know the function's mathematical **definition**, which describes exactly what the function does. Starting from the function's definition, you can use your existing math skills to find the function's domain, its range, and its inverse function. You must also know the **graph** of the function; what the function looks like if you plot x versus $f(x)$ in the Cartesian plane (page 29). It's also a good idea to remember the **values** of the function for some important inputs. Finally—and this is the part that takes time—you must learn about the function's **relations** to other functions.

Definitions

A *function* is a mathematical object that takes numbers as inputs and gives numbers as outputs. We use the notation

$$f\colon A \to B$$

to denote a function from the input set A to the output set B. In this book, we mostly study functions that take real numbers as inputs and give real numbers as outputs: $f\colon \mathbb{R} \to \mathbb{R}$.

We now define some fancy technical terms used to describe the input and output sets.

- The *domain* of a function is the set of allowed input values.
- The *image* or *range* of the function f is the set of all possible output values of the function.
- The *codomain* of a function describes the type of outputs the function has.

To illustrate the subtle difference between the image of a function and its codomain, consider the function $f(x) = x^2$. The quadratic function is of the form $f\colon \mathbb{R} \to \mathbb{R}$. The function's domain is $\mathbb{R}$ (it takes real numbers as inputs) and its codomain is $\mathbb{R}$ (the outputs are real numbers too), however, not all outputs are possible. The *image* of the function $f(x) = x^2$ consists only of the nonnegative real numbers $[0, \infty \equiv \{y \in \mathbb{R} \mid y \geq 0\}$.

A function is not a number; rather, it is a *mapping* from numbers to numbers. For any input x, the output value of f for that input is denoted $f(x)$.

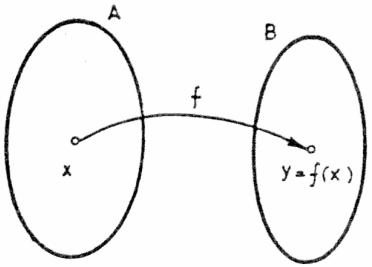


Figure 1.5: An abstract representation of a function f from the set A to the set B. The function f is the arrow which *maps* each input x in A to an output $f(x)$ in B. The output of the function $f(x)$ is also denoted y.

We say "f maps x to $f(x)$," and use the following terminology to classify the type of mapping that a function performs:

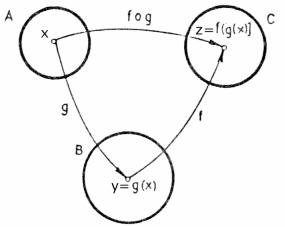
- A function is *one-to-one* or *injective* if it maps different inputs to different outputs.
- A function is *onto* or *surjective* if it covers the entire output set (in other words, if the image of the function is equal to the function's codomain).
- A function is *bijective* if it is both injective and surjective. In this case, f is a *one-to-one correspondence* between the input set and the output set: for each of the possible outputs $y \in Y$ (surjective part), there exists exactly one input $x \in X$, such that $f(x) = y$ (injective part).

The term *injective* is an allusion from the 1940s inviting us to picture the actions of injective functions as pipes through which numbers flow like fluids. Since a fluid cannot be compressed, the output space must be at least as large as the input space. A modern synonym for injective functions is to say they are *two-to-two*. If we imagine two specks of paint floating around in the "input fluid," an injective function will contain two distinct specks of paint in the "output fluid." In contrast, non-injective functions can map several different inputs to the same output. For example $f(x) = x^2$ is not injective since the inputs 2 and -2 are both mapped to the output value 4.

Function composition

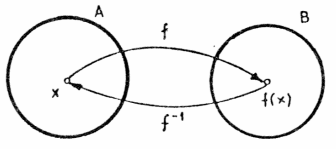
We can combine two simple functions by chaining them together to build a more complicated function. This act of applying one function after another is called *function composition*. Consider for example the composition:

$$f \circ g\,(x) \equiv f(\,g(x)\,) = z.$$



The diagram on the right illustrates what is going on. First, the function $g : A \to B$ acts on some input x to produce an intermediary value $y = g(x)$ in the set B. The intermediary value y is then passed through the function $f : B \to C$ to produce the final output value $z = f(y) = f(g(x))$ in the set C. We can think of the *composite function* $f \circ g$ as a function in its own right. The function $f \circ g : A \to C$ is defined through the formula $f \circ g\,(x) \equiv f(g(x))$.

Inverse function



Recall that a *bijective* function is a one-to-one correspondence between a set of input values and a set of output values. Given a bijective function $f : A \to B$, there exists an inverse function $f^{-1} : B \to A$, which performs the *inverse mapping* of f. If you start from some x, apply f, and then apply f^{-1}, you'll arrive—full circle—back to the original input x:

$$f^{-1}\big(\, f(x) \,\big) \equiv f^{-1} \circ f \,(x) = x.$$

This inverse function is represented abstractly as a backward arrow, that puts the value $f(x)$ back to the x it came from.

Function names

We use short symbols like $+$, $-$, $\times$, and $\div$ to denote most of the important functions used in everyday life. We also use the weird *surd* notation to denote n^{th} root $\sqrt[n]{\ }$ and superscripts to denote exponents. All other functions are identified and denoted by their *name*. If I want to compute the *cosine* of the angle $60°$ (a function describing the ratio between the length of one side of a right-angle triangle and the hypotenuse), I write $\cos(60°)$, which means I want the value of the cos function for the input $60°$.

Incidentally, the function cos has a nice output value for that specific angle: $\cos(60°) \equiv \frac{1}{2}$. Therefore, seeing $\cos(60°)$ somewhere in an equation is the same as seeing $\frac{1}{2}$. To find other values of the function, say $\cos(33.13°)$, you'll need a calculator. A scientific calculator features a convenient little $\boxed{\cos}$ button for this very purpose.

Handles on functions

When you learn about functions you learn about the different "handles" by which you can "grab" these mathematical objects. The main handle for a function is its **definition**: it tells you the precise way to calculate the output when you know the input. The function definition is an important handle, but it is also important to "feel" what the function does intuitively. How does one get a feel for a function?

Table of values

One simple way to represent a function is to look at a list of input-output pairs: $\big\{\{\text{in} = x_1, \text{out} = f(x_1)\}, \{\text{in} = x_2, \text{out} = f(x_2)\}, \{\text{in} = x_3, \text{out} = f(x_3)\}, \ldots\big\}$. A more compact notation for the input-output pairs is $\{(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), \ldots\}$. You can

make your own little **table of values**, pick some random inputs, and record the output of the function in the second column:

$$\begin{array}{rcl} \text{input} = x & \to & f(x) = \text{output} \\ 0 & \to & f(0) \\ 1 & \to & f(1) \\ 55 & \to & f(55) \\ x_4 & \to & f(x_4). \end{array}$$

In addition to choosing random numbers for your table, it's also generally a good idea to check the function's values at $x = 0$, $x = 1$, $x = 100$, $x = -1$, and any other important-looking x value.

Function graph

One of the best ways to feel a function is to look at its graph. A graph is a line on a piece of paper that passes through all input-output pairs of a function. Imagine you have a piece of paper, and on it you draw a blank *coordinate system* as in Figure 1.6.

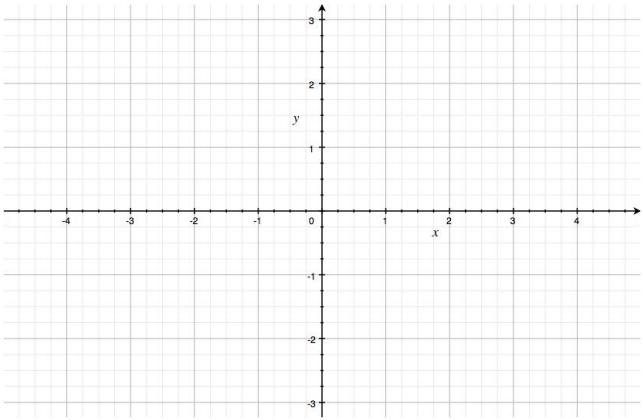


Figure 1.6: An empty (x, y)-coordinate system that you can use to plot the graph of *any* function $f(x)$. The graph of $f(x)$ consists of all the points for which $(x, y) = (x, f(x))$. See Figure 1.4 on page 31 for the graph of $f(x) = x^2$.

The horizontal axis, sometimes called the *abscissa*, is used to measure x. The vertical axis is used to measure $f(x)$. Because writing out $f(x)$ every time is long and tedious, we use a short, single-letter alias to denote the output value of f as follows:

$$y \equiv f(x) = \text{output}.$$

www.ebook3000.com

Think of each input-output pair of the function f as a point (x, y) in the coordinate system. The graph of a function is a representational drawing of everything the function does. If you understand how to interpret this drawing, you can infer everything there is to know about the function.

Facts and properties

Another way to feel a function is by knowing the function's properties. This approach boils down to learning facts about the function and its relation to other functions. An example of a mathematical fact is $\sin(30^\circ) = \frac{1}{2}$. An example of a mathematical relation is the equation $\sin^2 x + \cos^2 x = 1$, which indicates a link between the sin function and the cos function.

The more you know about a function, the more "paths" your brain builds to connect to that function. Real math knowledge is not memorization; it requires establishing a graph of associations between different areas of information in your brain. Each concept is a *node* in this graph, and each fact you know about this concept is an *edge*. Mathematical thought is the usage of this graph to produce calculations and mathematical arguments called proofs. For example, by connecting your knowledge of the fact $\sin(30^\circ) = \frac{1}{2}$ with the relation $\sin^2 x + \cos^2 x = 1$, you can show that $\cos(30^\circ) = \frac{\sqrt{3}}{2}$. Note the notation $\sin^2(x)$ means $(\sin(x))^2$.

To develop mathematical skills, it is vital to practice path-building between related concepts by solving exercises and reading and writing mathematical proofs. With this book, I will introduce you to many paths between concepts; it's up to you to reinforce these by using what you've learned to solve problems.

Example

Consider the function f from the real numbers to the real numbers ($f\colon \mathbb{R} \to \mathbb{R}$) defined by the quadratic expression

$$f(x) = x^2 + 2x + 3.$$

The value of f when $x = 1$ is $f(1) = 1^2 + 2(1) + 3 = 1 + 2 + 3 = 6$. When $x = 2$, the output is $f(2) = 2^2 + 2(2) + 3 = 4 + 4 + 3 = 11$. What is the value of f when $x = 0$?

Example 2

Consider the exponential function with base 2:

$$f(x) = 2^x.$$

This function is crucial to computer systems. For instance, RAM memory chips come in powers of two because the memory space is exponential in the number of "address lines" used on the chip. When $x = 1$, $f(1) = 2^1 = 2$. When x is 2 we have $f(2) = 2^2 = 4$. The function is therefore described by the following input-output pairs: $(0, 1)$, $(1, 2)$, $(2, 4)$, $(3, 8)$, $(4, 16)$, $(5, 32)$, $(6, 64)$, $(7, 128)$, $(8, 256)$, $(9, 512)$, $(10, 1024)$, $(11, 2048)$, $(12, 4096)$, etc. Recall that any number raised to exponent 0 gives 1. Thus, the exponential function passes through the point $(0, 1)$. Recall also that negative exponents lead to fractions: $(-1, \frac{1}{2^1} = \frac{1}{2}), (-2, \frac{1}{2^2} = \frac{1}{4}), (-3, \frac{1}{2^3} = \frac{1}{8})$, etc.

Discussion

In this section we talked a lot about functions in general, but we haven't said much about any function specifically. There are many useful functions out there, and we can't discuss them all here. In the next section, we'll introduce 10 functions of strategic importance for all of science. If you get a grip on these functions, you'll be able to understand all of physics and calculus and handle *any* problem your teacher may throw at you.

1.9 Function reference

Your *function vocabulary* determines how well you can express yourself mathematically in the same way that your English vocabulary determines how well you can express yourself in English. The following pages aim to embiggen your function vocabulary so you won't be caught with your pants down when the teacher tries to pull some trick on you at the final.

If you are seeing these functions for the first time, don't worry about remembering all the facts and properties on the first reading. We will use these functions throughout the rest of the book so you will have plenty of time to become familiar with them. Just remember to come back to this section if you ever get stuck on a function.

Line

The equation of a line describes an input-output relationship where the change in the output is *proportional* to the change in the input. The equation of a line is

$$f(x) = mx + b.$$

The constant m describes the slope of the line. The constant b is called the y-intercept and it corresponds to the value of the function

when $x = 0$.

Graph

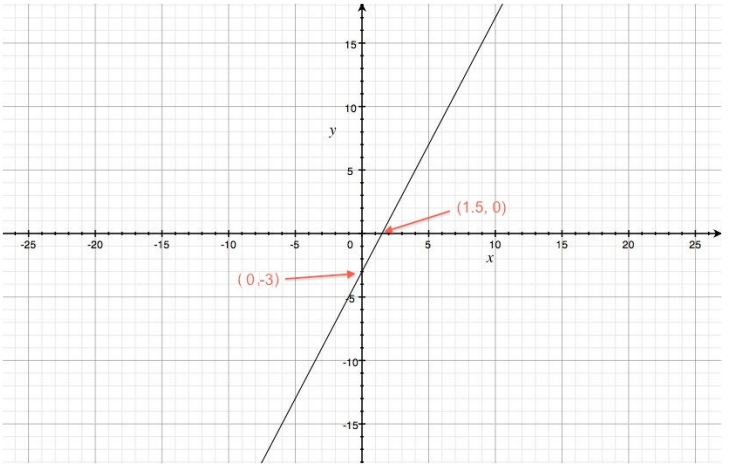


Figure 1.7: The graph of the function $f(x) = 2x - 3$. The slope is $m = 2$. The y-intercept of this line is at $y = -3$. The x-intercept is at $x = \frac{3}{2}$.

Properties

- Domain: $x \in \mathbb{R}$.
  The function $f(x) = mx+b$ is defined for all input values $x \in \mathbb{R}$.
- Image: $x \in \mathbb{R}$ if $m \neq 0$. If $m = 0$ the function is constant $f(x) = b$, so the image set contains only a single number $\{b\}$.
- b/m: the x-intercept of $f(x) = mx + b$. The x-intercept is obtained by solving $f(x) = 0$.
- A unique line passes through any two points (x_1, y_1) and (x_2, y_2) if $x_1 \neq x_2$.
- The inverse to the line $f(x) = mx + b$ is $f^{-1}(x) = \frac{1}{m}(x - b)$, which is also a line.

General equation

A line can also be described in a more symmetric form as a relation:

$$Ax + By = C.$$

This is known as the *general* equation of a line. The general equation for the line shown in Figure 1.7 is $2x - 1y = 3$.

Given the general equation of a line $Ax+By = C$, you can convert to the function form $y = f(x) = mx + b$ using $b = \frac{C}{B}$ and $m = \frac{-A}{B}$.

Square

The function x *squared*, is also called the *quadratic* function, or *parabola*. The formula for the quadratic function is

$$f(x) = x^2.$$

The name "quadratic" comes from the Latin *quadratus* for square, since the expression for the area of a square with side length x is x^2.

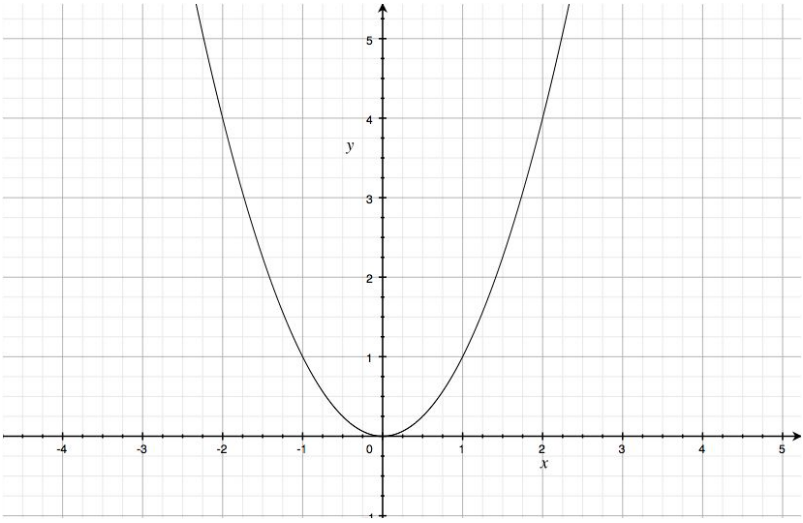


Figure 1.8: Plot of the quadratic function $f(x) = x^2$. The graph of the function passes through the following (x, y) coordinates: $(-2, 4)$, $(-1, 1)$, $(0, 0)$, $(1, 1)$, $(2, 4)$, $(3, 9)$, etc.

Properties

- Domain: $x \in \mathbb{R}$.
  The function $f(x) = x^2$ is defined for all input values $x \in \mathbb{R}$.
- Image: $f(x) \in [0, \infty$.
  The outputs are never negative: $x^2 \geq 0$, for all $x \in \mathbb{R}$.
- The function x^2 is the inverse of the square root function $\sqrt{x}$.
- $f(x) = x^2$ is *two-to-one*: it sends both x and $-x$ to the same output value $x^2 = (-x)^2$.
- The quadratic function is *convex*, meaning it curves upward.

Square root

The square root function is denoted

$$f(x) = \sqrt{x} \equiv x^{\frac{1}{2}}.$$

The square root $\sqrt{x}$ is the inverse function of the square function x^2 for $x \geq 0$. The symbol $\sqrt{c}$ refers to the *positive* solution of $x^2 = c$. Note that $-\sqrt{c}$ is also a solution of $x^2 = c$.

Graph

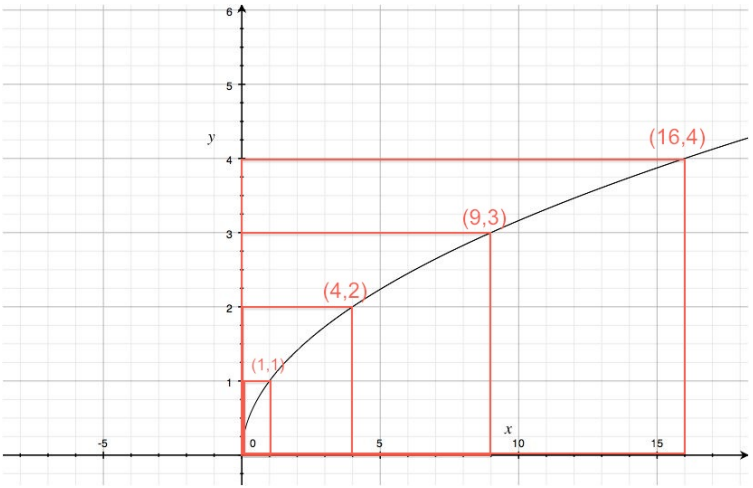


Figure 1.9: The graph of the function $f(x) = \sqrt{x}$. The domain of the function is $x \in [0, \infty$. You can't take the square root of a negative number.

Properties

- Domain: $x \in [0, \infty$.
  The function $f(x) = \sqrt{x}$ is only defined for nonnegative inputs $x \geq 0$. There is no real number y such that y^2 is negative, hence the function $f(x) = \sqrt{x}$ is not defined for negative inputs x.
- Image: $f(x) \in [0, \infty$.
  The outputs of the function $f(x) = \sqrt{x}$ are never negative: $\sqrt{x} \geq 0$, for all $x \in [0, \infty$.

In addition to *square* root, there is also *cube* root $f(x) = \sqrt[3]{x} \equiv x^{\frac{1}{3}}$, which is the inverse function for the cubic function $f(x) = x^3$. We have $\sqrt[3]{8} = 2$ since $2 \times 2 \times 2 = 8$. More generally, we can define the n^{th}-root function $\sqrt[n]{x}$ as the inverse function of x^n.

Absolute value

The absolute value function tells us the *size* of numbers without paying attention to whether the number is positive or negative. We can compute a number's absolute value by *ignoring the sign* of the input number. Thus, a number's absolute value corresponds to its distance from the origin of the number line.

Another way of thinking about the absolute value function is to say it multiplies negative numbers by -1 to "cancel" their negative sign:

$$f(x) = |x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

Graph

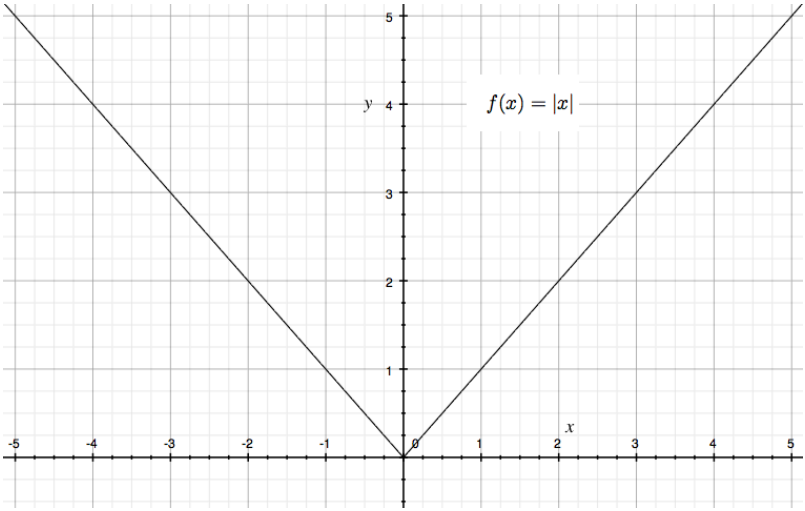


Figure 1.10: The graph of the absolute value function $f(x) = |x|$.

Properties

- Always returns a non-negative number
- The combination of squaring followed by square-root is equivalent to the absolute value function:

  $$\sqrt{x^2} \equiv |x|,$$

  since squaring destroys the sign.

Polynomial functions

The general equation for a polynomial function of degree n is written,

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n.$$

The constants a_i are known as the *coefficients* of the polynomial.

Parameters

- n: the *degree* of the polynomial
- a_0: the constant term
- a_1: the *linear* coefficient, or first-order coefficient
- a_2: the *quadratic* coefficient
- a_3: the *cubic* coefficient
- a_n: the n^{th} order coefficient

A polynomial of degree n has $n + 1$ coefficients: $a_0, a_1, a_2, \ldots, a_n$.

Properties

- Domain: $x \in \mathbb{R}$. Polynomials are defined for all inputs $x \in \mathbb{R}$.
- Image: depends on the coefficients
- The sum of two polynomials is also a polynomial.

Even and odd functions

The polynomials form an entire family of functions. Depending on the choice of degree n and coefficients a_0, a_1, ..., a_n, a polynomial function can take on many different shapes. We'll study polynomials and their properties in more detail in Section 1.10, but for now consider the following observations about the symmetries of polynomials:

- If a polynomial contains only even powers of x, like $f(x) = 1 + x^2 - x^4$ for example, we call this polynomial *even*. Even polynomials have the property $f(x) = f(-x)$. The sign of the input doesn't matter.
- If a polynomial contains only odd powers of x, for example $g(x) = x + x^3 - x^9$, we call this polynomial *odd*. Odd polynomials have the property $g(x) = -g(-x)$.
- If a polynomial has both even and odd terms then it is neither even nor odd.

The terminology of *odd* and *even* applies to functions in general and not just to polynomials. All functions that satisfy $f(x) = f(-x)$ are called *even functions*, and all functions that satisfy $f(x) = -f(-x)$ are called *odd functions*.

Sine

The sine function represents a fundamental unit of vibration. The graph of $\sin(x)$ *oscillates* up and down and crosses the x-axis multiple times. The shape of the graph of $\sin(x)$ corresponds to the shape of a vibrating string. See Figure 1.11.

In the remainder of this book, we'll meet the function $\sin(x)$ many times. We will define the function $\sin(x)$ more formally as a trigonometric ratio in Section 1.11 (page 58). In Chapter 2 we will use $\sin(x)$ and $\cos(x)$ (another trigonometric ratio) to work out the *components* of vectors. Later in Chapter **??**, we will learn how the sine function can be used to describe waves and periodic motion.

At this point in the book, however, we don't want to go into too much detail about all these applications. Let's hold off the discussion about vectors, triangles, angles, and ratios of lengths of sides and instead just focus on the graph of the function $f(x) = \sin(x)$.

Graph

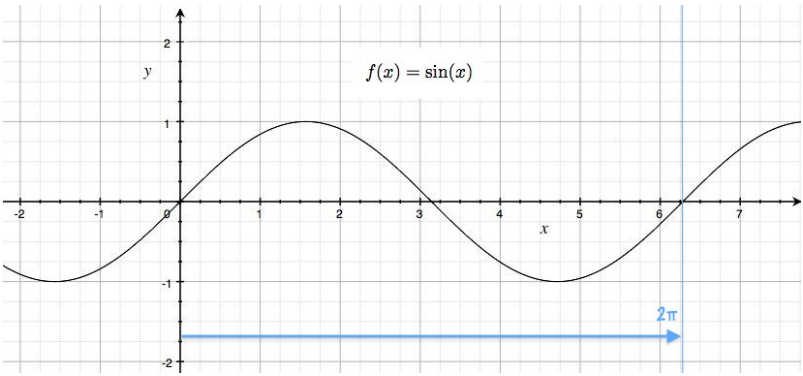
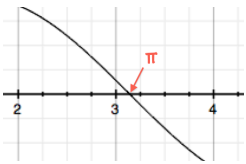


Figure 1.11: The graph of the function $y = \sin(x)$ passes through the following (x, y) coordinates: $(0, 0)$, $(\frac{\pi}{6}, \frac{1}{2})$, $(\frac{\pi}{4}, \frac{\sqrt{2}}{2})$, $(\frac{\pi}{3}, \frac{\sqrt{3}}{2})$, $(\frac{\pi}{2}, 1)$, $(\frac{2\pi}{3}, \frac{\sqrt{3}}{2})$, $(\frac{3\pi}{4}, \frac{\sqrt{2}}{2})$, $(\frac{5\pi}{6}, \frac{1}{2})$, and $(\pi, 0)$. For $x \in [\pi, 2\pi]$ the function has the same shape as for $x \in [0, \pi]$ but with negative values.

Let's start at $x = 0$ and follow the graph of the function $\sin(x)$ as it goes up and down. The graph starts from $(0, 0)$ and smoothly increases until it reaches the maximum value at $x = \frac{\pi}{2}$. Afterward, the function comes back down to cross the x-axis at $x = \pi$. After π, the function drops below the x-axis and reaches its



minimum value of -1 at $x = \frac{3\pi}{2}$. It then travels up again to cross the x-axis at $x = 2\pi$. This 2π-long cycle repeats after $x = 2\pi$. This is why we call the function *periodic*—the shape of the graph repeats.

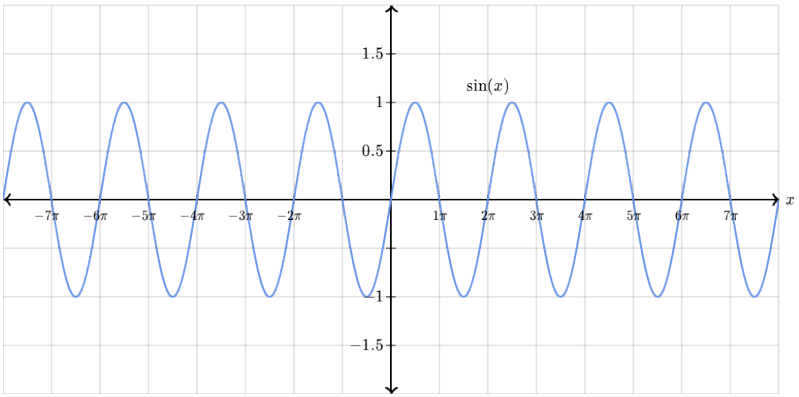


Figure 1.12: The graph of $\sin(x)$ from $x = 0$ to $x = 2\pi$ repeats periodically everywhere else on the number line.

Properties

- Domain: $x \in \mathbb{R}$.
  The function $f(x) = \sin(x)$ is defined for all input values $x \in \mathbb{R}$.
- Image: $\sin(x) \in [-1, 1]$.
  The outputs of the sine function are always between -1 and 1.
- Roots: $[\ldots, -3\pi, -2\pi, -\pi, 0, \pi, 2\pi, 3\pi, \ldots]$.
  The function $\sin(x)$ has roots at all multiples of π.
- The function is periodic, with period 2π: $\sin(x) = \sin(x + 2\pi)$.
- The sin function is *odd*: $\sin(x) = -\sin(-x)$.
- Relation to cos: $\sin^2 x + \cos^2 x = 1$
- Relation to csc: $\csc(x) \equiv \frac{1}{\sin x}$ (csc is read *cosecant*)
- The inverse function of $\sin(x)$ is denoted as $\sin^{-1}(x)$, not to be confused with $(\sin(x))^{-1} = \frac{1}{\sin(x)} \equiv \csc(x)$. Sometimes the function $\sin^{-1}(x)$ is denoted "arcsin(x)."
- The number $\sin(\theta)$ is the length-ratio of the vertical side and the hypotenuse in a right-angle triangle with angle θ at the base.

Links

[See the Wikipedia page for nice illustrations]
`http://en.wikipedia.org/wiki/Sine`

Cosine

The cosine function is the same as the sine function *shifted* by $\frac{\pi}{2}$ to the left: $\cos(x) = \sin(x + \frac{\pi}{2})$. Thus everything you know about the sine function also applies to the cosine function.

Graph

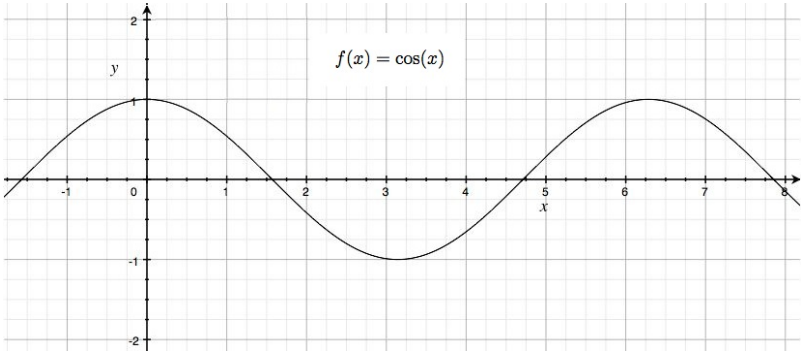


Figure 1.13: The graph of the function $y = \cos(x)$ passes through the following (x, y) coordinates: $(0, 1)$, $(\frac{\pi}{6}, \frac{\sqrt{3}}{2})$, $(\frac{\pi}{4}, \frac{\sqrt{2}}{2})$, $(\frac{\pi}{3}, \frac{1}{2})$, $(\frac{\pi}{2}, 0)$, $(\frac{2\pi}{3}, -\frac{1}{2})$, $(\frac{3\pi}{4}, -\frac{\sqrt{2}}{2})$, $(\frac{5\pi}{6}, -\frac{\sqrt{3}}{2})$, and $(\pi, -1)$.

The cos function starts at $\cos(0) = 1$, then drops down to cross the x-axis at $x = \frac{\pi}{2}$. Cos continues until it reaches its minimum value at $x = \pi$. The function then moves upward, crossing the x-axis again at $x = \frac{3\pi}{2}$, and reaching its maximum value at $x = 2\pi$.

Properties

- Domain: $x \in \mathbb{R}$
- Image: $\cos(x) \in [-1, 1]$
- Roots: $[\ldots, -\frac{3\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{2}, \ldots]$.
- Relation to sin: $\sin^2 x + \cos^2 x = 1$
- Relation to sec: $\sec(x) \equiv \frac{1}{\cos x}$ (sec is read *secant*)
- The inverse function of $\cos(x)$ is denoted $\cos^{-1}(x)$.
- The cos function is *even*: $\cos(x) = \cos(-x)$.
- The number $\cos(\theta)$ is the length-ratio of the horizontal side and the hypotenuse in a right-angle triangle with angle θ at the base.

Tangent

The tangent function is the ratio of the sine and cosine functions:

$$f(x) = \tan(x) \equiv \frac{\sin(x)}{\cos(x)}.$$

Graph

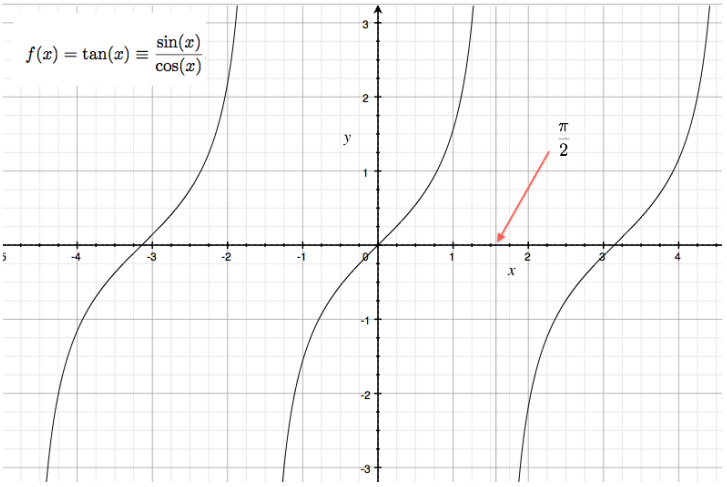


Figure 1.14: The graph of the function $f(x) = \tan(x)$.

Properties

- Domain: $\{x \in \mathbb{R} \mid x \neq \frac{(2n+1)\pi}{2} \text{ for any } n \in \mathbb{Z}\}$.
- Range: $x \in \mathbb{R}$.
- The function tan is periodic with period π.
- The tan function "blows up" at values of x where $\cos x = 0$. These are called *asymptotes* of the function and their locations are $x = \ldots, \frac{-3\pi}{2}, \frac{-\pi}{2}, \frac{\pi}{2}, \frac{3\pi}{2}, \ldots$.
- Value at $x = 0$: $\tan(0) = \frac{0}{1} = 0$, because $\sin(0) = 0$.
- Value at $x = \frac{\pi}{4}$: $\tan\left(\frac{\pi}{4}\right) = \frac{\sin(\frac{\pi}{4})}{\cos(\frac{\pi}{4})} = \frac{\frac{\sqrt{2}}{2}}{\frac{\sqrt{2}}{2}} = 1$.
- The number $\tan(\theta)$ is the length-ratio of the vertical and the horizontal sides in a right-angle triangle with angle θ.
- The inverse function of $\tan(x)$ is $\tan^{-1}(x)$.
- The inverse tangent function is used to compute the angle at the base in a right-angle triangle with horizontal side length ℓ_h and vertical side length ℓ_v: $\theta = \tan^{-1}\left(\frac{\ell_v}{\ell_h}\right)$.

Exponential

The exponential function base $e = 2.7182818\ldots$ is denoted

$$f(x) = e^x \equiv \exp(x).$$

Graph

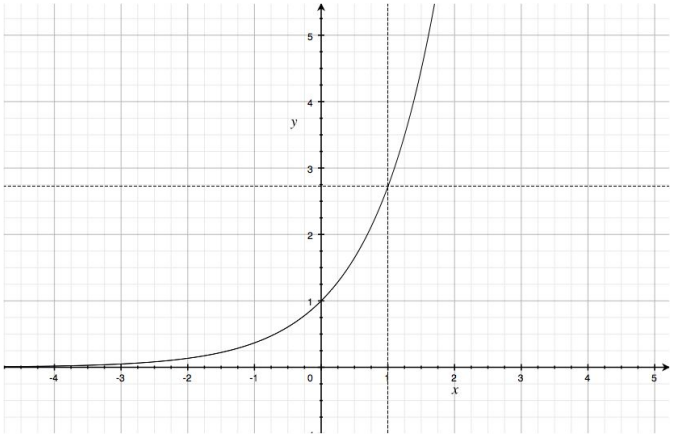


Figure 1.15: The graph of the exponential function $f(x) = e^x$ passes through the following (x, y) coordinates: $(-2, \frac{1}{e^2})$, $(-1, \frac{1}{e})$, $(0, 1)$, $(1, e)$, $(2, e^2)$, $(3, e^3 = 20.08\ldots)$, $(5, 148.41\ldots)$, and $(10, 22026.46\ldots)$.

Properties

- Domain: $x \in \mathbb{R}$
- Range: $e^x \in (0, \infty$
- $f(a)f(b) = f(a + b)$ since $e^a e^b = e^{a+b}$.
- The derivative (the slope of the graph) of the exponential function is the exponential function: $f(x) = e^x \quad \Rightarrow \quad f'(x) = e^x$.

A more general exponential function would be $f(x) = Ae^{\gamma x}$, where A is the initial value, and γ (the Greek letter *gamma*) is the *rate* of the exponential. For $\gamma > 0$, the function $f(x)$ is increasing, as in Figure 1.15. For $\gamma < 0$, the function is decreasing and tends to zero for large values of x. The case $\gamma = 0$ is special since $e^0 = 1$, so $f(x)$ is a constant of $f(x) = A1^x = A$.

Links

[The exponential function 2^x evaluated]
`http://www.youtube.com/watch?v=e4MSN6IImpI`

Natural logarithm

The natural logarithm function is denoted

$$f(x) = \ln(x) = \log_e(x).$$

The function $\ln(x)$ is the inverse function of the exponential e^x.

Graph

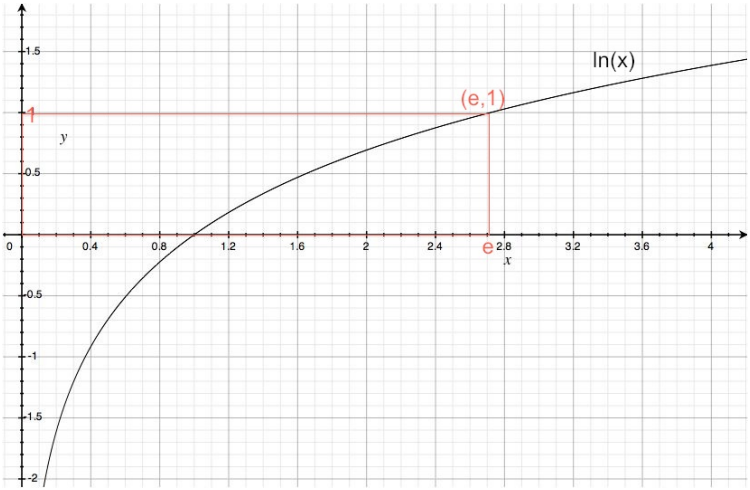


Figure 1.16: The graph of the function $\ln(x)$ passes through the following (x, y) coordinates: $(\frac{1}{e^2}, -2)$, $(\frac{1}{e}, -1)$, $(1, 0)$, $(e, 1)$, $(e^2, 2)$, $(e^3, 3)$, $(148.41\ldots, 5)$, and $(22026.46\ldots, 10)$.

Function transformations

Often, we're asked to adjust the shape of a function by scaling it or moving it, so that it passes through certain points. For example, if we wanted to make a function g with the same shape as the absolute value function $f(x) = |x|$, but for which $g(0) = 3$, we would use the function $g(x) = |x| + 3$.

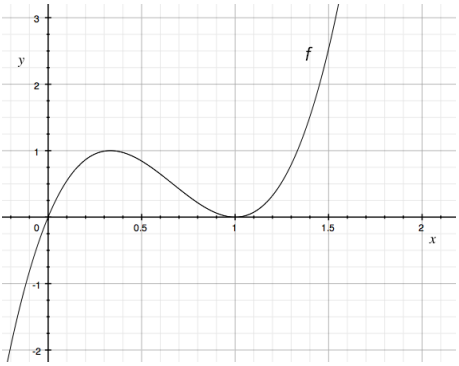
In this section, we'll discuss the four basic transformations you can perform on *any* function f to obtain a transformed function g:

- Vertical translation: $g(x) = f(x) + k$
- Horizontal translation: $g(x) = f(x - h)$
- Vertical scaling: $g(x) = Af(x)$
- Horizontal scaling: $g(x) = f(ax)$

By applying these transformations, we can *move* and *stretch* a generic function to give it any desired shape.

The next couple of pages illustrate all of the above transformations on the function

$$f(x) = 6.75(x^3 - 2x^2 + x).$$



We'll work with this function because it has distinctive features in both the horizontal and vertical directions. By observing this function's graph, we see its x-intercepts are at $x = 0$ and $x = 1$. We can confirm this mathematically by factoring the expression:

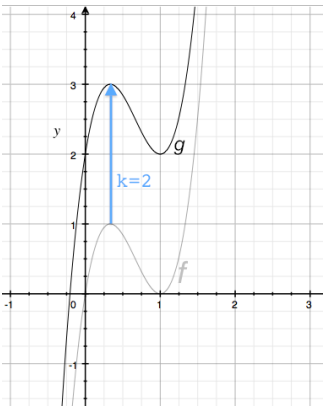
$$f(x) = 6.75x(x^2 - 2x + 1) = 6.75x(x - 1)^2.$$

The function $f(x)$ also has a local maximum at $x = \frac{1}{3}$, and the value of the function at that maximum is $f(\frac{1}{3}) = 1$.

Vertical translations

To move a function $f(x)$ *up* by k units, add k to the function:

$$g(x) = f(x) + k.$$



The function $g(x)$ will have exactly the same shape as $f(x)$, but it will be *translated* (the mathematical term for moved) upward by k units.

Recall the function $f(x) = 6.75(x^3 - 2x^2 + x)$. To move the function up by $k = 2$ units, we can write

$$g(x) = f(x) + 2 = 6.75(x^3 - 2x^2 + x) + 2,$$

and the graph of $g(x)$ will be as it is shown to the right. Recall the original function $f(x)$ crosses the x-axis at $x = 0$. The transformed function $g(x)$ has the property $g(0) = 2$. The maximum at $x = \frac{1}{3}$ has similarly shifted in value from $f(\frac{1}{3}) = 1$ to $g(\frac{1}{3}) = 3$.

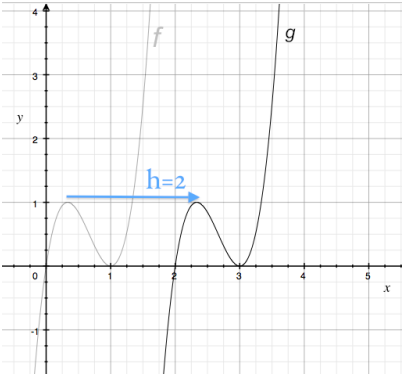
Horizontal translation

We can move a function f to the right by h units by *subtracting* h from x and using $(x - h)$ as the function's input argument:

$$g(x) = f(x - h).$$

The point $(0, f(0))$ on $f(x)$ now corresponds to the point $(h, g(h))$ on $g(x)$.

The graph to the right shows the function $f(x) = 6.75(x^3 - 2x^2 + x)$, as well as the function $g(x)$, which is shifted to the right by $h = 2$ units:



$$g(x) = f(x - 2) = 6.75\left[(x - 2)^3 - 2(x - 2)^2 + (x - 2)\right].$$

The original function f gives us $f(0) = 0$ and $f(1) = 0$, so the new function $g(x)$ must give $g(2) = 0$ and $g(3) = 0$. The maximum at $x = \frac{1}{3}$ has similarly shifted by two units to the right, $g(2 + \frac{1}{3}) = 1$.

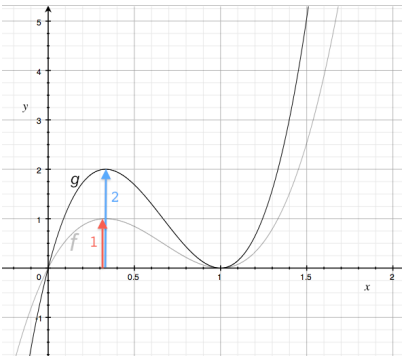
Vertical scaling

To stretch or compress the shape of a function vertically, we can multiply it by some constant A and obtain

$$g(x) = Af(x).$$

If $|A| > 1$, the function will be stretched. If $|A| < 1$, the function will be compressed. If A is negative, the function will flip upside down, which is a *reflection* through the x-axis.

There is an important difference between vertical translation and vertical scaling. Translation moves all points of the function by the same amount, whereas scaling moves each point proportionally to that point's distance from the x-axis.



The function $f(x) = 6.75(x^3 - 2x^2 + x)$, when stretched vertically by a factor of $A = 2$, becomes the function

$$g(x) = 2f(x) = 13.5(x^3 - 2x^2 + x).$$

The x-intercepts $f(0) = 0$ and $f(1) = 0$ do not move, and remain at $g(0) = 0$ and $g(1) = 0$. The maximum at $x = \frac{1}{3}$ has doubled in value as $g(\frac{1}{3}) = 2$. Indeed, all values of $f(x)$ have been stretched upward by a factor of 2, as we can verify using the point $f(1.5) = 2.5$, which has become $g(1.5) = 5$.

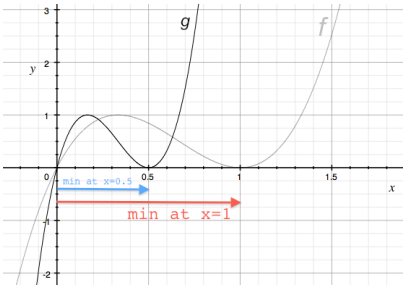
Horizontal scaling

To stretch or compress a function horizontally, we can multiply the input value by some constant a to obtain:

$$g(x) = f(ax).$$

If $|a| > 1$, the function will be compressed. If $|a| < 1$, the function will be stretched. Note that the behaviour here is the opposite of vertical scaling. If a is a negative number, the function will also flip horizontally, which is a reflection through the y-axis.

The graph on the right shows $f(x) = 6.75(x^3 - 2x^2 + x)$, as well as the function $g(x)$, which is $f(x)$ compressed horizontally by a factor of $a = 2$:



$$\begin{aligned} g(x) &= f(2x) \\ &= 6.75\big[(2x)^3 - 2(2x)^2 + (2x)\big]. \end{aligned}$$

The x-intercept $f(0) = 0$ does not move since it is on the y-axis. The x-intercept $f(1) = 0$ does move, however, and we have $g(0.5) = 0$. The maximum at $x = \frac{1}{3}$ moves to $g(\frac{1}{6}) = 1$. All values of $f(x)$ are compressed toward the y-axis by a factor of 2.

General quadratic function

The general quadratic function takes the form

$$f(x) = A(x - h)^2 + k,$$

where x is the input, and A, h, and k are the *parameters*.

Parameters

- A: the slope multiplier
  - ▷ The larger the absolute value of A, the steeper the slope.
  - ▷ If $A < 0$ (negative), the function opens downward.
- h: the horizontal displacement of the function. Notice that subtracting a number inside the bracket $(\ )^2$ (positive h) makes the function go to the right.
- k: the vertical displacement of the function

Graph

The graph in Figure 1.17 illustrates a quadratic function with parameters $A = 1$, $h = 1$ (one unit shifted to the right), and $k = -2$ (two units shifted down).

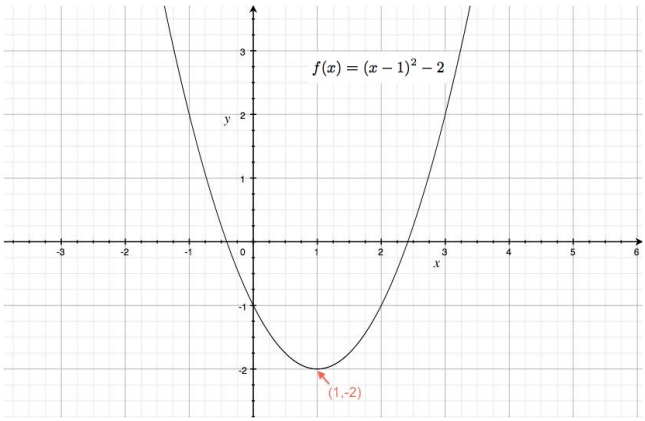


Figure 1.17: The graph of the function $f(x) = (x-1)^2 - 2$ is the same as the basic function $f(x) = x^2$, but shifted one unit to the right and two units down.

If a quadratic crosses the x-axis, it can be written in factored form:

$$f(x) = A(x-a)(x-b),$$

where a and b are the two roots. Another common way of writing a quadratic function is $f(x) = Ax^2 + Bx + C$.

Properties

- There is a unique quadratic function that passes through any three points (x_1, y_1), (x_2, y_2) and (x_3, y_3), if the points have different x-coordinates: $x_1 \neq x_2$, $x_2 \neq x_3$, and $x_1 \neq x_3$.
- The derivative of $f(x) = Ax^2 + Bx + C$ is $f'(x) = 2Ax + B$.

General sine function

Introducing all possible parameters into the sine function gives us:

$$f(x) = A\sin\left(\tfrac{2\pi}{\lambda}x - \phi\right),$$

where A, λ, and ϕ are the function's parameters.

Parameters

- A: the amplitude describes the distance above and below the x-axis that the function reaches as it oscillates.
- λ: the *wavelength* of the function:

$$\lambda \equiv \{ \text{ the distance from one peak to the next } \}.$$

- ϕ: is a phase shift, analogous to the horizontal shift h, which we have seen. This number dictates where the oscillation starts. The default sine function has zero phase shift ($\phi = 0$), so it passes through the origin with an increasing slope.

The "bare" sin function $f(x) = \sin(x)$ has wavelength 2π and produces outputs that oscillate between -1 and $+1$. When we multiply the bare function by the constant A, the oscillations will range between $-A$ and A. When the input x is scaled by the factor $\frac{2\pi}{\lambda}$, the wavelength of the function becomes λ.

1.10 Polynomials

The polynomials are a simple and useful family of functions. For example, quadratic polynomials of the form $f(x) = ax^2 + bx + c$ often arise when describing physics phenomena.

Definitions

- x: the variable
- $f(x)$: the polynomial. We sometimes denote polynomials $P(x)$ to distinguish them from generic function $f(x)$.
- Degree of $f(x)$: the largest power of x that appears in the polynomial
- Roots of $f(x)$: the values of x for which $f(x) = 0$

The most general first-degree polynomial is a line $f(x) = mx + b$, where m and b are arbitrary constants. The most general second-degree polynomial is $f(x) = a_2x^2 + a_1x + a_0$, where again a_0, a_1, and

a_2 are arbitrary constants. We call a_k the *coefficient* of x^k, since this is the number that appears in front of x^k. Following the pattern, a third-degree polynomial will look like $f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$.

In general, a polynomial of degree n has the equation

$$f(x) = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0.$$

Or, if we use *summation* notation, we can write the polynomial as

$$f(x) = \sum_{k=0}^{n} a_kx^k.$$

The symbol Σ (the Greek letter *sigma*) stands for *summation*.

Solving polynomial equations

Very often in math, you will have to *solve* polynomial equations of the form

$$A(x) = B(x),$$

where $A(x)$ and $B(x)$ are both polynomials. Recall from earlier that to *solve*, we must find the value of x that makes the equality true.

Say the revenue of your company is a function of the number of products sold x, and can be expressed as $R(x) = 2x^2 + 2x$. Say also the cost you incur to produce x objects is $C(x) = x^2 + 5x + 10$. You want to determine the amount of product you need to produce to break even, that is, so that revenue equals cost: $R(x) = C(x)$. To find the break-even value x, solve the equation

$$2x^2 + 2x = x^2 + 5x + 10.$$

This may seem complicated since there are xs all over the place. No worries! We can turn the equation into its "standard form," and then use the quadratic formula. First, move all the terms to one side until only zero remains on the other side:

$$\begin{aligned}
2x^2 + 2x \quad &- x^2 = x^2 + 5x + 10 \quad - x^2 \\
x^2 + 2x \quad &- 5x = 5x + 10 \quad - 5x \\
x^2 - 3x \quad &- 10 = 10 \quad - 10 \\
x^2 - 3x - 10 &= 0.
\end{aligned}$$

Remember, if we perform the same operations on both sides of the equation, the equation remains true. Therefore, the values of x that satisfy

$$x^2 - 3x - 10 = 0,$$

namely $x = -2$ and $x = 5$, also satisfy

$$2x^2 + 2x = x^2 + 5x + 10,$$

which is the original problem we're trying to solve.

This "shuffling of terms" approach will work for any polynomial equation $A(x) = B(x)$. We can always rewrite it as $C(x) = 0$, where $C(x)$ is a new polynomial with coefficients equal to the difference of the coefficients of A and B. Don't worry about which side you move all the coefficients to because $C(x) = 0$ and $0 = -C(x)$ have exactly the same solutions. Furthermore, the degree of the polynomial C can be no greater than that of A or B.

The form $C(x) = 0$ is the *standard form* of a polynomial, and we'll explore several formulas you can use to find its solution(s).

Formulas

The formula for solving the polynomial equation $P(x) = 0$ depends on the *degree* of the polynomial in question.

First

For a first-degree polynomial equation, $P_1(x) = mx + b = 0$, the solution is $x = \frac{-b}{m}$: just move b to the other side and divide by m.

Second

For a second-degree polynomial,

$$P_2(x) = ax^2 + bx + c = 0,$$

the solutions are $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ and $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$.

If $b^2 - 4ac < 0$, the solutions will involve taking the square root of a negative number. In those cases, we say no real solutions exist.

Higher degrees

There is also a formula for polynomials of degree 3, but it is complicated. For polynomials with order ≥ 5, there does not exist a general analytical solution.

Using a computer

When solving real-world problems, you'll often run into much more complicated equations. To find the solutions of anything more complicated than the quadratic equation, I recommend using a computer algebra system like `SymPy`: `http://live.sympy.org`.

To make the computer solve the equation $x^2 - 3x + 2 = 0$ for you, type in the following:

```
>>> solve( x**2 - 3*x +2, x)          # usage: solve(expr, var)
[1, 2]
```

The function `solve` will find the roots of any equation of the form `expr = 0`. Indeed, we can verify that $x^2 - 3x + 2 = (x - 1)(x - 2)$, so $x = 1$ and $x = 2$ are the two roots.

Substitution trick

Sometimes you can solve fourth-degree polynomials by using the quadratic formula. Say you're asked to solve for x in

$$g(x) = x^4 - 3x^2 - 10 = 0.$$

Imagine this problem is on your exam, where you are not allowed the use of a computer. How does the teacher expect you to solve for x? The trick is to substitute $y = x^2$ and rewrite the same equation as

$$g(y) = y^2 - 3y - 10 = 0,$$

which you can solve by applying the quadratic formula. If you obtain the solutions $y = \alpha$ and $y = \beta$, then the solutions to the original fourth-degree polynomial are $x = \sqrt{\alpha}$ and $x = \sqrt{\beta}$, since $y = x^2$.

Since we're not taking an exam right now, we are allowed to use the computer to find the roots:

```
>>> solve(y**2 - 3*y -10, y)
[-2, 5]
>>> solve(x**4 - 3*x**2 -10 , x)
[sqrt(2)i, -sqrt(2)i, -sqrt(5) , sqrt(5) ]
```

Note how the second-degree polynomial has two roots, while the fourth-degree polynomial has four roots (two of which are imaginary, since we had to take the square root of a negative number to obtain them). The imaginary roots contain the unit imaginary number $i \equiv \sqrt{-1}$.

If you see this kind of problem on an exam, you should report the two real solutions as your answer—in this case $-\sqrt{5}$ and $\sqrt{5}$—without mentioning the imaginary solutions because you are not supposed to know about imaginary numbers yet. If you feel impatient and are ready to know about the imaginary numbers right now, feel free to skip ahead to the section on complex numbers (page 102).

1.11 Trigonometry

We can put any three lines together to make a triangle. What's more, if one of the triangle's angles is equal to 90°, we call this triangle a *right-angle triangle*.

In this section we'll discuss right-angle triangles in great detail and get to know their properties. We'll learn some fancy new terms like *hypotenuse*, *opposite*, and *adjacent*, which are used to refer to the different sides of a triangle. We'll also use the functions *sine*, *cosine*, and *tangent* to compute the *ratios of lengths* in right triangles.

Understanding triangles and their associated trigonometric functions is of fundamental importance: you'll need this knowledge for your future understanding of mathematical subjects like vectors and complex numbers, as well as physics subjects like oscillations and waves.

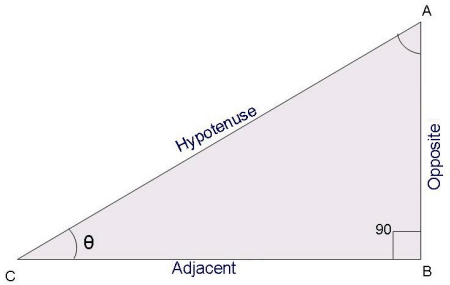


Figure 1.18: A right-angle triangle. The angle θ and the names of the sides of the triangle are indicated.

Concepts

- A, B, C: the three *vertices* of the triangle
- θ: the angle at the vertex C. Angles can be measured in degrees or radians.
- opp $\equiv \overline{AB}$: the length of the *opposite* side to θ
- adj $\equiv \overline{BC}$: the length of side *adjacent* to θ
- hyp $\equiv \overline{AC}$: the *hypotenuse*. This is the triangle's longest side.
- h: the "height" of the triangle (in this case $h = \text{opp} = \overline{AB}$)
- $\sin\theta \equiv \frac{\text{opp}}{\text{hyp}}$: the *sine* of theta is the ratio of the length of the opposite side and the length of hypotenuse.
- $\cos\theta \equiv \frac{\text{adj}}{\text{hyp}}$: the *cosine* of theta is the ratio of the adjacent length and the hypotenuse length.
- $\tan\theta \equiv \frac{\sin\theta}{\cos\theta} \equiv \frac{\text{opp}}{\text{adj}}$: the *tangent* is the ratio of the opposite length divided by the adjacent length.

Pythagoras' theorem

In a right-angle triangle, the length of the hypotenuse squared is equal to the sum of the squares of the lengths of the other sides:

$$|\text{adj}|^2 + |\text{opp}|^2 = |\text{hyp}|^2.$$

If we divide both sides of the above equation by $|\text{hyp}|^2$, we obtain

$$\frac{|\text{adj}|^2}{|\text{hyp}|^2} + \frac{|\text{opp}|^2}{|\text{hyp}|^2} = 1,$$

which can be rewritten as

$$\cos^2\theta \ + \sin^2\theta = 1.$$

This is a powerful *trigonometric identity* that describes an important relationship between sin and cos.

Sin and cos

Meet the trigonometric functions, or trigs for short. These are your new friends. Don't be shy now, say hello to them.

"Hello."

"Hi."

"Soooooo, you are like functions right?"

"Yep," sin and cos reply in chorus.

"Okay, so what do you do?"

"Who me?" asks cos. "Well I tell the ratio... hmm... Wait, are you asking what I do as a *function* or specifically what *I* do?"

"Both I guess?"

"Well, as a function, I take angles as inputs and I give ratios as answers. More specifically, I tell you how 'wide' a triangle with that angle will be," says cos all in one breath.

"What do you mean wide?" you ask.

"Oh yeah, I forgot to say, the triangle must have a hypotenuse of length 1. What happens is there is a point P that moves around on a circle of radius 1, and we *imagine* a triangle formed by the point P, the origin, and the point on the x-axis located directly below the point P."

"I am not sure I get it," you confess.

"Let me try explaining," says sin. "Look on the next page, and you'll see a circle. This is the unit circle because it has a radius of 1. You see it, yes?"

"Yes."

"This circle is really cool. Imagine a point P that starts from the point $P(0) = (1, 0)$ and moves along the circle of radius 1. The x and

y coordinates of the point $P(\theta) = (P_x(\theta),\ P_y(\theta))$ as a function of θ are

$$P(\theta) = (P_x(\theta),\ P_y(\theta)) = (\cos\theta,\ \sin\theta).$$

So, *either* you can think of us in the context of triangles, or you think of us in the context of the unit circle."

"Cool. I kind of get it. Thanks so much," you say, but in reality you are weirded out. Talking functions? "Well guys. It was nice to meet you, but I have to get going, to finish the rest of the book."

"See you later," says cos.

"Peace out," says sin.

The unit circle

The unit circle consists of all points (x, y) that satisfy the equation $x^2 + y^2 = 1$. A point $P = (P_x, P_y)$ on the unit circle has coordinates $(P_x, P_y) = (\cos\theta, \sin\theta)$, where θ is the angle P makes with the x-axis.

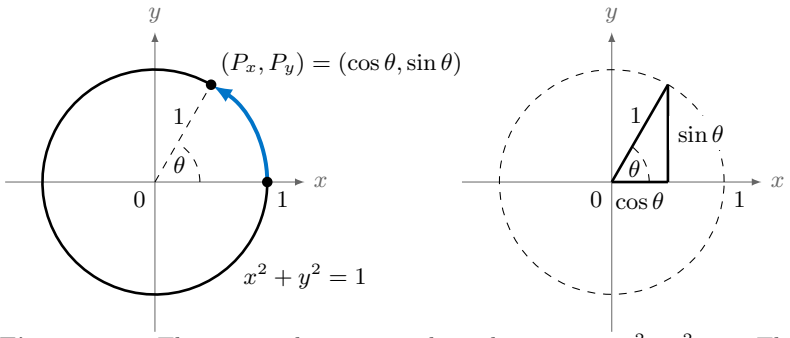


Figure 1.19: The unit circle corresponds to the equation $x^2+y^2 = 1$. The coordinates of the point P on the unit circle are $P_x = \cos\theta$ and $P_y = \sin\theta$.

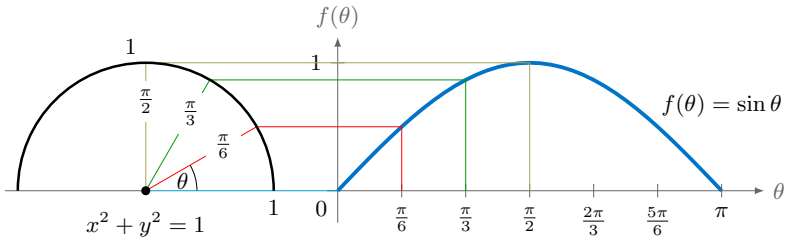


Figure 1.20: The function $f(\theta) = \sin\theta$ describes the vertical position of a point P that travels along the unit circle. The first half of a cycle is shown.

You should be familiar with the values of sin and cos for all angles that are multiples of $\frac{\pi}{6}$ (30°) or $\frac{\pi}{4}$ (45°). All of them are shown in

Figure 1.21. For each angle, the x-coordinate (the first number in the bracket) is $\cos\theta$, and the y-coordinate is $\sin\theta$.

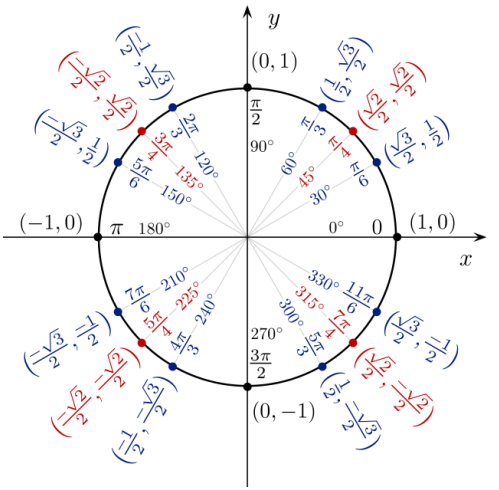


Figure 1.21: The unit circle. The coordinates of the point on the unit circle $(\cos\theta, \sin\theta)$ are indicated for several important values of the angle θ.

Maybe you're thinking that's way too much to remember. Don't worry, you just have to memorize one fact:

$$\sin(30°) = \sin\left(\frac{\pi}{6}\right) = \frac{1}{2}\,.$$

Knowing this, you can determine all the other angles. Let's start with $\cos(30°)$. We know that at 30°, point P on the unit circle has the vertical coordinate $\frac{1}{2} = \sin(30°)$. We also know the cos quantity we are looking for is, by definition, the horizontal component:

$$P = (\cos(30°), \sin(30°)).$$

Key fact: all points on the unit circle are a distance of 1 from the origin. Knowing that P is a point on the unit circle, and knowing the value of $\sin(30°)$, we can solve for $\cos(30°)$. Start with the following identity,

$$\cos^2\theta \ + \sin^2\theta = 1,$$

which is true for *all* angles θ. Moving things around, we obtain

$$\cos(30°) = \sqrt{1 - \sin^2(30°)} = \sqrt{1 - \frac{1}{4}} = \sqrt{\frac{3}{4}} = \frac{\sqrt{3}}{2}\,.$$

To find the values of $\cos(60°)$ and $\sin(60°)$, observe the symmetry of the circle. 60 degrees measured from the x-axis is the same as 30

degrees measured from the y-axis. From this, we know $\cos(60°) = \sin(30°) = \frac{1}{2}$. Therefore, $\sin(60°) = \frac{\sqrt{3}}{2}$.

To find the values of sin and cos for angles that are multiples of 45°, we need to find the value a such that

$$a^2 + a^2 = 1,$$

since at 45°, the horizontal and vertical coordinates will be the same. Solving for a we find $a = \frac{1}{\sqrt{2}}$, but people don't like to see square roots in the denominator, so we write

$$\frac{\sqrt{2}}{2} = \cos(45°) = \sin(45°).$$

All other angles in the circle behave like the three angles above, with one difference: one or more of their components has a negative sign. For example, 150° is just like 30°, except its x component is negative. Don't memorize all the values of sin and cos; if you ever need to determine their values, draw a little circle and use the symmetry of the circle to find the sin and cos components.

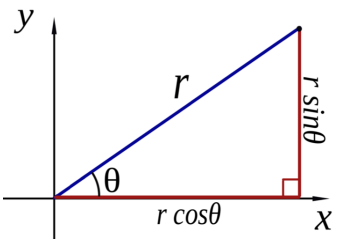
Non-unit circles

Consider a point $Q(\theta)$ at an angle of θ on a circle with radius $r \neq 1$. How can we find the x and y-coordinates of the point $Q(\theta)$?

We saw that the coefficients $\cos\theta$ and $\sin\theta$ correspond to the x and y-coordinates of a point on the *unit* circle ($r = 1$). To obtain the coordinates for a point on a circle of radius r, we must *scale* the coordinates by a factor of r:

$$Q(\theta) = (Q_x(\theta), Q_y(\theta)) = (r\cos\theta, r\sin\theta).$$

The take-away message is that you can use the functions $\cos\theta$ and $\sin\theta$ to find the "horizontal" and "vertical" components of any length r.



From this point on in the book, we'll always talk about the length of the *adjacent* side as $r_x = r\cos\theta$, and the length of the *opposite* side as $r_y = r\sin\theta$. It is extremely important you get comfortable with this notation.

The reasoning behind the above calculations is as follows:

$$\cos\theta \equiv \frac{\text{adj}}{\text{hyp}} = \frac{r_x}{r} \quad \Rightarrow \quad r_x = r\cos\theta,$$

and

$$\sin\theta \equiv \frac{\text{opp}}{\text{hyp}} = \frac{r_y}{r} \quad \Rightarrow \quad r_y = r\sin\theta.$$

Calculators

Make sure to set your calculator to the correct units for working with angles. What should you type into your calculator to compute the sine of 30 degrees? If your calculator is set to degrees, simply type: `30`, `sin`, `=`.

If your calculator is set to radians, you have two options:

1. Change the `mode` of the calculator so it works in degrees.
2. Convert 30° to radians

$$30\ [^\circ] \times \frac{2\pi\ [\text{rad}]}{360\ [^\circ]} = \frac{\pi}{6}\ [\text{rad}],$$

and type: `π`, `/`, `6`, `sin`, `=` on your calculator.

Links

[Unit-circle walkthrough and tricks by patrickJMT on YouTube]
`bit.ly/1mQg9Cj` and `bit.ly/1hvA702`

1.12 Trigonometric identities

There are a number of important relationships between the values of the functions sin and cos. Here are three of these relationships, known as *trigonometric identities*. There about a dozen other identities that are less important, but you should memorize these three.

The three identities to remember are:

1. Unit hypotenuse

$$\sin^2(\theta) + \cos^2(\theta) = 1.$$

The unit hypotenuse identity is true by the Pythagoras theorem and the definitions of sin and cos. The sum of the squares of the sides of a triangle is equal to the square of the hypotenuse.

2. sico + sico

$$\sin(a+b) = \sin(a)\cos(b) + \sin(b)\cos(a).$$

The mnemonic for this identity is "sico + sico."

3. coco − sisi

$$\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b).$$

The mnemonic for this identity is "coco - sisi." The negative sign is there because it's not good to be a sissy.

Derived formulas

If you remember the above three formulas, you can derive pretty much all the other trigonometric identities.

Double angle formulas

Starting from the sico-sico identity as explained above, and setting $a = b = x$, we can derive the following identity:

$$\sin(2x) = 2\sin(x)\cos(x).$$

Starting from the coco-sisi identity, we obtain

$$\begin{aligned}\cos(2x) &= \cos^2(x) - \sin^2(x) \\ &= 2\cos^2(x) - 1 \;= 2\left(1 - \sin^2(x)\right) - 1 \;= 1 - 2\sin^2(x).\end{aligned}$$

The formulas for expressing $\sin(2x)$ and $\cos(2x)$ in terms of $\sin(x)$ and $\cos(x)$ are called *double angle formulas*.

If we rewrite the double-angle formula for $\cos(2x)$ to isolate the $\sin^2$ or the $\cos^2$ term, we obtain the *power-reduction formulas*:

$$\cos^2(x) = \frac{1}{2}\left(1 + \cos(2x)\right), \qquad \sin^2(x) = \frac{1}{2}\left(1 - \cos(2x)\right).$$

Self similarity

Sin and cos are periodic functions with period 2π. Adding a multiple of 2π to the function's input does not change the function:

$$\sin(x + 2\pi) = \sin(x + 124\pi) = \sin(x), \qquad \cos(x + 2\pi) = \cos(x).$$

Furthermore, sin and cos are self similar within each 2π cycle:

$$\sin(\pi - x) = \sin(x), \qquad \cos(\pi - x) = -\cos(x).$$

Sin is cos, cos is sin

It shouldn't be surprising if I tell you that sin and cos are actually $\frac{\pi}{2}$-shifted versions of each other:

$$\cos(x) = \sin\Big(x + \frac{\pi}{2}\Big) = \sin\Big(\frac{\pi}{2} - x\Big), \;\; \sin(x) = \cos\Big(x - \frac{\pi}{2}\Big) = \cos\Big(\frac{\pi}{2} - x\Big).$$

Sum formulas

$$\sin(a) + \sin(b) = 2\sin\left(\frac{1}{2}(a+b)\right)\cos\left(\frac{1}{2}(a-b)\right),$$

$$\sin(a) - \sin(b) = 2\sin\left(\frac{1}{2}(a-b)\right)\cos\left(\frac{1}{2}(a+b)\right),$$

$$\cos(a) + \cos(b) = 2\cos\left(\frac{1}{2}(a+b)\right)\cos\left(\frac{1}{2}(a-b)\right),$$

$$\cos(a) - \cos(b) = -2\sin\left(\frac{1}{2}(a+b)\right)\sin\left(\frac{1}{2}(a-b)\right).$$

Product formulas

$$\sin(a)\cos(b) = \frac{1}{2}(\sin(a+b) + \sin(a-b)),$$

$$\sin(a)\sin(b) = \frac{1}{2}(\cos(a-b) - \cos(a+b)),$$

$$\cos(a)\cos(b) = \frac{1}{2}(\cos(a-b) + \cos(a+b)).$$

Discussion

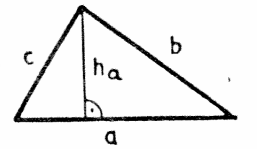
The above formulas will come in handy when you need to find some unknown in an equation, or when you are trying to simplify a trigonometric expression. I am not saying you should necessarily memorize them, but you should be aware that they exist.

1.13 Geometry

Triangles

The area of a triangle is equal to $\frac{1}{2}$ times the length of its base times its height:

$$A = \frac{1}{2}ah_a.$$

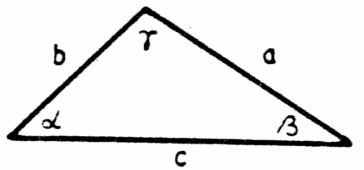


Note that h_a is the height of the triangle *relative to* the side a.

The perimeter of a triangle is

$$P = a + b + c.$$

Consider a triangle with internal angles α, β and γ. The sum of the inner angles in any triangle is equal to two right angles: $\alpha + \beta + \gamma = 180°$.



Sine rule The sine law is

$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)},$$

where α is the angle opposite to a, β is the angle opposite to b, and γ is the angle opposite to c.

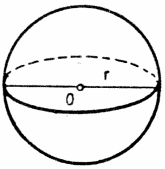
Cosine rule The cosine rules are

$$\begin{aligned} a^2 &= b^2 + c^2 - 2bc\cos(\alpha), \\ b^2 &= a^2 + c^2 - 2ac\cos(\beta), \\ c^2 &= a^2 + b^2 - 2ab\cos(\gamma). \end{aligned}$$

Sphere

A sphere is described by the equation

$$x^2 + y^2 + z^2 = r^2.$$



The surface area of a sphere is

$$A = 4\pi r^2,$$

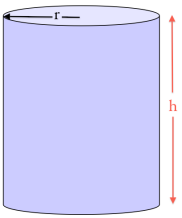
and the volume of a sphere is

$$V = \frac{4}{3}\pi r^3.$$

Cylinder

The surface area of a cylinder consists of the top and bottom circular surfaces, plus the area of the side of the cylinder:

$$A = 2\left(\pi r^2\right) + (2\pi r)h.$$



The formula for the volume of a cylinder is the product of the area of the cylinder's base times its height:

$$V = \left(\pi r^2\right) h.$$

www.ebook3000.com

Example You open the hood of your car and see 2.0 L written on top of the engine. The 2.0 L refers to the combined volume of the four pistons, which are cylindrical in shape. The owner's manual tells you the diameter of each piston (bore) is 87.5 mm, and the height of each piston (stroke) is 83.1 mm. Verify that the total volume of the cylinder displacement of your engine is indeed 1998789 $\text{mm}^3 \approx 2$ L.

1.14 Circle

The circle is a set of points located a constant distance from a centre point. This geometrical shape appears in many situations.

Definitions

- r: the radius of the circle
- A: the area of the circle
- C: the circumference of the circle
- (x, y): a point on the circle
- θ: the angle (measured from the x-axis) of some point on the circle

Formulas

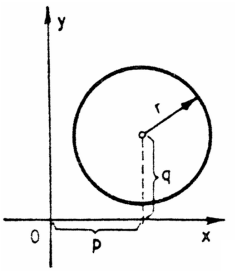
A circle with radius r centred at the origin is described by the equation

$$x^2 + y^2 = r^2.$$

All points (x, y) that satisfy this equation are part of the circle.

Rather than staying centred at the origin, the circle's centre can be located at any point (p, q) on the plane:

$$(x - p)^2 + (y - q)^2 = r^2.$$



Explicit function

The equation of a circle is a *relation* or an *implicit function* involving x and y. To obtain an *explicit function* $y = f(x)$ for the circle, we can solve for y to obtain

$$y = \sqrt{r^2 - x^2}, \quad -r \leq x \leq r,$$

and

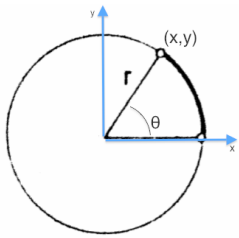
$$y = -\sqrt{r^2 - x^2}, \quad -r \leq x \leq r.$$

The explicit expression is really two functions, because a vertical line crosses the circle in two places. The first function corresponds to the top half of the circle, and the second function corresponds to the bottom half.

Polar coordinates

Circles are so common in mathematics that mathematicians developed a special "circular coordinate system" in order to describe them more easily.

It is possible to specify the coordinates (x, y) of any point on the circle in terms of the *polar coordinates* $r\angle\theta$, where r measures the distance of the point from the origin, and θ is the angle measured from the x-axis.



To convert from the polar coordinates $r\angle\theta$ to the (x, y) coordinates, use the trigonometric functions cos and sin:

$$x = r\cos\theta \qquad \text{and} \qquad y = r\sin\theta.$$

Parametric equation

We can describe *all* the points on the circle if we specify a fixed radius r and vary the angle θ over all angles: $\theta \in [0, 360°)$. A *parametric equation* specifies the coordinates $(x(\theta), y(\theta))$ for the points on a curve, for all values of the *parameter* θ. The parametric equation for a circle of radius r is given by

$$\{(x, y) \in \mathbb{R}^2 \mid x = r\cos\theta, y = r\sin\theta,\ \theta \in [0, 360°)\}.$$

Try to visualize the curve traced by the point $(x(\theta), y(\theta)) = (r\cos\theta, r\sin\theta)$ as θ varies from 0° to 360°. The point will trace out a circle of radius r.

If we let the parameter θ vary over a smaller interval, we'll obtain subsets of the circle. For example, the parametric equation for the top half of the circle is

$$\{(x, y) \in \mathbb{R}^2 \mid x = r\cos\theta, y = r\sin\theta,\ \theta \in [0, 180°]\}.$$

The top half of the circle is also described by $\{(x, y) \in \mathbb{R}^2 \mid y = \sqrt{r^2 - x^2},\ x \in [-r, r]\}$, where the parameter used is the x-coordinate.

Area

The area of a circle of radius r is $A = \pi r^2$.

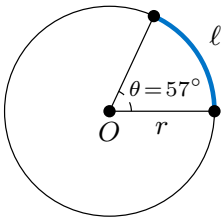
Circumference and arc length

The circumference of a circle is

$$C = 2\pi r.$$

This is the total length you can measure by following the curve all the way around to trace the outline of the entire circle.

What is the length of a part of the circle? Say you have a piece of the circle, called an *arc*, and that piece corresponds to the angle $\theta = 57°$. What is the arc's length ℓ?



If the circle's total length $C = 2\pi r$ represents a full 360° turn around the circle, then the arc length ℓ for a portion of the circle corresponding to the angle θ is

$$\ell = 2\pi r \frac{\theta}{360}\,.$$

The arc length ℓ depends on r, the angle θ, and a factor of $\frac{2\pi}{360}$.

Radians

Though degrees are commonly used as a measurement unit for angles, it's much better to measure angles in *radians*, since radians are the *natural* units for measuring angles. The conversion ratio between degrees and radians is

$$2\pi[\text{rad}] \;=\; 360°.$$

When measuring angles in radians, the arc length is given by:

$$\ell = r\theta_{\text{rad}}.$$

Measuring angles in radians is equivalent to measuring arc length on a circle with radius $r = 1$.

1.15 Solving systems of linear equations

We know that solving equations with one unknown—like $2x + 4 = 7x$, for instance—requires manipulating both sides of the equation until the unknown variable is *isolated* on one side. For this instance, we can subtract $2x$ from both sides of the equation to obtain $4 = 5x$, which simplifies to $x = \frac{4}{5}$.

What about the case when you are given two equations and must solve for two unknowns? For example,

$$x + 2y = 5,$$
$$3x + 9y = 21.$$

Can you find values of x and y that satisfy both equations?

Concepts

- x, y: the two unknowns in the equations
- $eq1, eq2$: a system of two equations that must be solved *simultaneously*. These equations will look like

$$a_1x + b_1y = c_1,$$
$$a_2x + b_2y = c_2,$$

  where as, bs, and cs are given constants.

Principles

If you have n equations and n unknowns, you can solve the equations simultaneously and find the values of the unknowns. There are several different approaches for solving equations simultaneously. We'll learn about three different approaches in this section.

Solution techniques

When solving for two unknowns in two equations, the best approach is to *eliminate* one of the variables from the equations. By combining the two equations appropriately, we can simplify the problem to the problem of finding one unknown in one equation.

Solving by substitution

We want to solve the following system of equations:

$$x + 2y = 5,$$
$$3x + 9y = 21.$$

We can isolate x in the first equation to obtain

$$x = 5 - 2y,$$
$$3x + 9y = 21.$$

Now *substitute* the expression for x from the top equation into the bottom equation:

$$3(5 - 2y) + 9y = 21.$$

We just eliminated one of the unknowns by substitution. Continuing, we expand the bracket to find

$$15 - 6y + 9y = 21,$$

or

$$3y = 6.$$

We find $y = 2$, but what is x? Easy. To solve for x, plug the value $y = 2$ into any of the equations we started from. Using the equation $x = 5 - 2y$, we find $x = 5 - 2(2) = 1$.

Solving by subtraction

Let's return to our set of equations to see another approach for solving:

$$\begin{aligned} x + 2y &= 5, \\ 3x + 9y &= 21. \end{aligned}$$

Observe that any equation will remain true if we multiply the whole equation by some constant. For example, we can multiply the first equation by 3 to obtain an equivalent set of equations:

$$\begin{aligned} 3x + 6y &= 15, \\ 3x + 9y &= 21. \end{aligned}$$

Why did I pick 3 as the multiplier? By choosing this constant, the x terms in both equations now have the same coefficient.

Subtracting two true equations yields another true equation. Let's subtract the top equation from the bottom one:

$$\cancel{3x} - \cancel{3x} + 9y - 6y = 21 - 15 \quad \Rightarrow \quad 3y = 6.$$

The $3x$ terms cancel. This subtraction eliminates the variable x because we multiplied the first equation by 3. We find $y = 2$. To find x, substitute $y = 2$ into one of the original equations:

$$x + 2(2) = 5,$$

from which we deduce that $x = 1$.

Solving by equating

There is a third way to solve the equations:

$$\begin{aligned} x + 2y &= 5, \\ 3x + 9y &= 21. \end{aligned}$$

We can isolate x in both equations by moving all other variables and constants to the right-hand sides of the equations:

$$\begin{aligned} x &= 5 - 2y, \\ x &= \frac{1}{3}(21 - 9y) = 7 - 3y. \end{aligned}$$

Though the variable x is unknown to us, we know two facts about it: x is equal to $5 - 2y$ and x is equal to $7 - 3y$. Therefore, we can eliminate x by equating the right-hand sides of the equations:

$$5 - 2y = 7 - 3y.$$

We solve for y by adding $3y$ to both sides and subtracting 5 from both sides. We find $y = 2$ then plug this value into the equation $x = 5 - 2y$ to find x. The solutions are $x = 1$ and $y = 2$.

Discussion

The three elimination techniques presented here will allow you to solve any system of n linear equations in n unknowns. Each time you perform a substitution, a subtraction, or an elimination by equating, you're simplifying the problem to a problem of finding $(n - 1)$ unknowns in a system of $(n - 1)$ equations. There is actually an entire course called linear algebra, in which you'll develop a more advanced, systematic approach for solving systems of linear equations.

Exercises

E1.1 Solve the system of equations simultaneously for x and y:

$$\begin{aligned} 2x + 4y &= 16, \\ 5x \ - \ y &= 7. \end{aligned}$$

E1.2 Solve the system of equations for the unknowns x, y, and z:

$$\begin{aligned} 2x + y - 4z &= 28, \\ x \ + y \ + \ z &= 8, \\ 2x - y - 6z &= 22. \end{aligned}$$

E1.3 Solve for p and q given the equations $p + q = 10$ and $p - q = 4$.

E1.4 Solve the following system of linear equations:

$$2x + 4y = 8,$$
$$4x + 3y = 11.$$

1.16 Set notation

A *set* is the mathematically precise notion for describing a group of objects. You don't need to know about sets to perform simple math; but more advanced topics require an understanding of what sets are and how to denote set membership and subset relations between sets.

Definitions

- *set*: a collection of mathematical objects
- S, T: the usual variable names for sets
- $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$: some important sets of numbers: the naturals, the integers, the rationals, and the real numbers, respectively.
- $\{$ definition $\}$: the curly brackets surround the definition of a set, and the expression inside the curly brackets describes what the set contains.

Set operations:

- $S \cup T$: the *union* of two sets. The union of S and T corresponds to the elements in either S or T.
- $S \cap T$: the *intersection* of the two sets. The intersection of S and T corresponds to the elements in both S and T.
- $S \setminus T$: *set minus*. The difference $S \setminus T$ corresponds to the elements of S that are not in T.

Set relations:

- $\subset$: is a subset of
- $\subseteq$: is a subset of or equal to

Special mathematical shorthand symbols and their corresponding meanings:

- $\forall$: for all
- $\exists$: there exists
- $\nexists$: there doesn't exist
- $|$: such that
- $\in$: element of
- $\notin$: not an element of

Sets

Much of math's power comes from *abstraction*: the ability to see the bigger picture and think *meta* thoughts about the common relationships between math objects. We can think of individual numbers like 3, -5, and π, or we can talk about the *set* of *all* numbers.

It is often useful to restrict our attention to a specific *subset* of the numbers as in the following examples.

Example 1: The nonnegative real numbers

Define $\mathbb{R}_+ \subset \mathbb{R}$ (read "$\mathbb{R}_+$ a subset of $\mathbb{R}$") to be the set of non-negative real numbers:

$$\mathbb{R}_+ \equiv \{\text{all } x \text{ in } \mathbb{R} \text{ such that } x \geq 0\},$$

or expressed more compactly,

$$\mathbb{R}_+ \equiv \{x \in \mathbb{R} \mid x \geq 0\}.$$

If we were to translate the above expression into plain English, it would read "the set $\mathbb{R}_+$ is defined as the set of all real numbers x such that x is greater or equal to zero."

Example 2: Odd and even integers

Define the set of even integers as

$$E \equiv \{n \in \mathbb{Z} \mid \tfrac{n}{2} \in \mathbb{Z}\} = \{\ldots, -2, 0, 2, 4, 6, \ldots\}$$

and the set of odd integers as

$$O \equiv \{n \in \mathbb{Z} \mid \tfrac{n+1}{2} \in \mathbb{Z}\} = \{\ldots, -3, -1, 1, 3, 5, \ldots\}.$$

In both of the above examples, we use the mathematical notation $\{\ldots \mid \ldots\}$ to define the sets. Inside the curly brackets we first describe the general kind of objects we are talking about, followed by the symbol "|" (read "such that"), followed by the conditions that must be satisfied by all elements of the set.

Number sets

Recall how we defined the fundamental sets of numbers in the beginning of this chapter. It is worthwhile to review them briefly.

The *natural* numbers form the set derived when you start from 0 and add 1 any number of times:

$$\mathbb{N} \equiv \{0, 1, 2, 3, 4, 5, 6, \ldots\}.$$

The integers are the numbers derived by adding or subtracting 1 some number of times:

$$\mathbb{Z} \equiv \{x \mid x = \pm n, n \in \mathbb{N}\}.$$

When we allow for divisions between integers, we get the rational numbers:

$$\mathbb{Q} \equiv \left\{ z \;\middle|\; z = \frac{x}{y} \text{ where } x \text{ and } y \text{ are in } \mathbb{Z}, \text{ and } y \neq 0 \right\}.$$

The broader class of real numbers also includes all rationals as well as irrational numbers like $\sqrt{2}$ and π:

$$\mathbb{R} \equiv \{\pi, e, -1.53929411\ldots,\ 4.99401940129401\ldots,\ \ldots\}.$$

Finally, we have the set of complex numbers:

$$\mathbb{C} \equiv \{1, i, 1+i, 2+3i, \ldots\}.$$

Note that the definitions of $\mathbb{R}$ and $\mathbb{C}$ are not very precise. Rather than giving a precise definition of each set inside the curly brackets as we did for $\mathbb{Z}$ and $\mathbb{Q}$, we instead stated some examples of the elements in the set. Mathematicians sometimes do this and expect you to guess the general pattern for all the elements in the set.

The following inclusion relationship holds for the fundamental sets of numbers:

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}.$$

This relationship means every natural number is also an integer. Every integer is a rational number. Every rational number is a real. Every real number is also a complex number.

New vocabulary

The specialized notation used by mathematicians can be difficult to get used to. You must learn how to read symbols like $\exists$, $\subset$, $|$, and $\in$ and translate their meaning in the sentence. Indeed, learning advanced mathematics notation is akin to learning a new language.

To help you practice the new vocabulary, we will look at an ancient mathematical proof and express it in terms of modern mathematical symbols.

Square-root of 2 is irrational

Claim: $\sqrt{2} \notin \mathbb{Q}$. This means there do not exist numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$. The last sentence expressed in mathematical notation would read,

$$\nexists\, m \in \mathbb{Z}, n \in \mathbb{Z} \;\mid\; m/n = \sqrt{2}.$$

To prove the claim we'll use a technique called *proof by contradiction*. We begin by assuming the opposite of what we want to prove: that there exist numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$. We'll then carry out some simple algebra steps and in the end we'll obtain an equation that is not true—we'll arrive at a contradiction. Arriving at a contradiction means our original supposition is wrong: there are no numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$.

Proof: Suppose there exist numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$. We can assume the integers m and n have no common factors. In particular, m and n cannot both be even, otherwise they would both contain at least one factor of 2. Next, we'll investigate whether m is an even number $m \in E$, or an odd number $m \in O$. Look back to Example 2 for the definitions of the sets O and E.

Before we check for even and oddness, it will help to point out the fact that the action of squaring an integer preserves its odd/even nature. An even number times an even number gives an even number: if $e \in E$ then $e^2 \in E$. Similarly, an odd number times an odd number gives an odd number: if $o \in O$ then $o^2 \in O$.

We proceed with the proof. We assume $m/n = \sqrt{2}$. Taking the square of both sides of this equation, we obtain

$$\frac{m^2}{n^2} = 2 \qquad \Rightarrow \qquad m^2 = 2n^2.$$

If we analyze the last equation in more detail, we can conclude that m cannot be an odd number, or written "$m \notin O$" in math. If m is an odd number then m^2 will also be odd, but this would contradict the above equation since the right-hand side of the equation contains the factor 2 and every number containing a factor 2 is even, not odd. If m is an integer ($m \in \mathbb{Z}$) and m is not odd ($m \notin O$) then it must be that m is even ($m \in E$).

If m is even, then it contains a factor of 2, so it can be written as $m = 2q$ where q is some other number $q \in \mathbb{Z}$. The exact value of q is not important. Let's revisit the equation $m^2 = 2n^2$ once more, this time substituting $m = 2q$ into the equation:

$$(2q)^2 = 2n^2 \quad \Rightarrow \quad 2q^2 = n^2.$$

By a similar reasoning as before, we can conclude n cannot be odd ($n \notin O$) so n must be even ($n \in E$). However, this statement contradicts our initial assumption that m and n do not have any common factors!

The fact that we arrived at a contradiction means we must have made a mistake somewhere in our reasoning. Since each of the steps we carried out were correct, the mistake must be in the original

premise, namely that "there exist numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$." Rather, the opposite must be true: "there do not exist numbers $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$ such that $m/n = \sqrt{2}$." The last statement is equivalent to saying $\sqrt{2}$ is irrational, which is what we wanted to prove. □

Set relations and operations

Figure 1.22 illustrates the notion of a set B that is strictly contained in the set A. We say $B \subset A$ if $\forall b \in B$, we also have $b \in A$, and $\exists a \in A$ such that $a \notin B$. In other words, we write $B \subset A$ whenever the set A contains B, but there exists at least one element in A that is not an element of B.

Also illustrated in Figure 1.22 is the union of two sets $A \cup B$, which includes all the elements of both A and B. If $e \in A \cup B$, then $e \in A$ and/or $e \in B$.

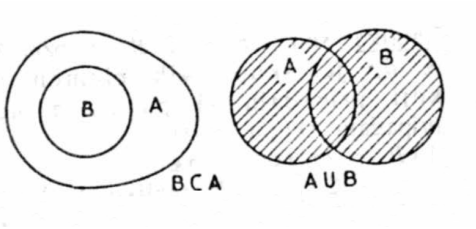


Figure 1.22: The left side of the figure is an illustration of a set B which is strictly contained in another set A, denoted $B \subset A$. The right side of the figure illustrates the union of two sets $A \cup B$.

The set intersection $A \cap B$ and set minus $A \setminus B$ are illustrated in Figure 1.23.

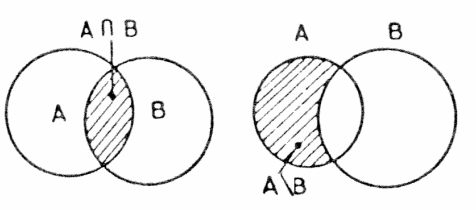


Figure 1.23: The left side of the figure shows the intersection between the sets $A \cap B$. The intersection of two sets contains the elements that are part of both sets. The right side of the figure shows the set difference $A \setminus B$, which consists of all the elements that are in A but not in B.

Sets related to functions

A function that takes real variables as inputs and produces real numbers as outputs is denoted $f : \mathbb{R} \to \mathbb{R}$. The *domain* of a function is the set of all possible inputs to the function that produce an output:

$$\text{Dom}(f) \equiv \{x \in \mathbb{R} \mid f(x) \in \mathbb{R}\}.$$

Inputs for which the function is undefined are not part of the domain. For instance the function $f(x) = \sqrt{x}$ is not defined for negative inputs, so we have $\text{Dom}(f) = \mathbb{R}_+$.

The *image set* of a function is the set of all possible outputs of the function:

$$\text{Im}(f) \equiv \{y \in \mathbb{R} \mid \exists x \in \mathbb{R},\ y = f(x)\}.$$

For example, the function $f(x) = x^2$ has the image set $\text{Im}(f) = \mathbb{R}_+$ since the outputs it produces are always non-negative.

Discussion

Knowledge of the precise mathematical jargon introduced in this section is not crucial to understanding the rest of this book. That being said, I wanted to expose you to it here because this is the language in which mathematicians think. Most advanced math textbooks will assume you understand technical mathematical notation.

1.17 Math problems

We've now reached the first section of problems in this book. The purpose of these problems is to give you a way to comprehensively practice your math fundamentals. In the real world, you'll rarely have to solve equations by hand, however, knowing how to solve math equations and manipulate math expressions will be very useful in later chapters of this book. At times, honing your math chops might seem like tough mental work, but at the end of each problem, you'll gain a stronger foothold on all the subjects you've been learning about. You'll also experience a small *achievement buzz* after each problem you vanquish.

I have a special message to readers who are learning math just for fun: you can either try the problems in this section or skip them. Since you have no upcoming exam on this material, you could skip ahead to the next chapter without any immediate consequences. However (and it's a big however), those readers who don't take a crack at these problems will be missing a significant opportunity.

Sit down to do them later today, or another time when you're properly caffeinated. If you take the initiative to make time for math, you'll find yourself developing lasting comprehension and true math fluency. Without the practice of solving problems, however, you're extremely likely to forget most of what you've learned within a month, simple as that. You'll still remember the big ideas, but the details will be fuzzy and faded. Don't break the pace now: with math, it's very much *use it or lose it*!

By solving some of the problems in this section, you'll remember a lot more stuff. Make sure you step away from the pixels while you're solving problems. You don't need fancy technology to do math; grab a pen and some paper from the printer and you'll be fine. Do yourself a favour: put your phone in airplane-mode, close the lid of your laptop, and move away from desktop computers. Give yourself some time to think. Yes, I know you can look up the answer to any question in five seconds on the Internet, and you can use `live.sympy.org` to solve any math problem, but that is like outsourcing the thinking. Descartes, Leibniz, and Riemann did most of their work with pen and paper and they did well. Spend some time with math the way the masters did.

P1.1 Solve for x in the equation $x^2 - 9 = 7$.

P1.2 Solve for x in the equation $\cos^{-1}\left(\frac{x}{A}\right) - \phi = \omega t$.

P1.3 Solve for x in the equation $\frac{1}{x} = \frac{1}{a} + \frac{1}{b}$.

P1.4 Use a calculator to find the values of the following expressions:

(1) $\sqrt[4]{3^3}$ (2) 2^{10} (3) $7^{\frac{1}{4}} - 10$ (4) $\frac{1}{2}\ln(e^{22})$

P1.5 Compute the following expressions involving fractions:

(1) $\frac{1}{2} + \frac{1}{4}$ (2) $\frac{4}{7} - \frac{23}{5}$ (3) $1\frac{3}{4} + 1\frac{31}{32}$

P1.6 Use the basic rules of algebra to simplify the following expressions:

(1) $ab\,\frac{1}{a}\,b^2cb^{-3}$ (2) $\frac{abc}{bca}$ (3) $\frac{27a^2}{\sqrt{9abba}}$

(4) $\frac{a(b+c)-ca}{b}$ (5) $\frac{a}{c\sqrt[3]{b}}\frac{b^{\frac{4}{3}}}{a^2}$ (6) $(x+a)(x+b) - x(a+b)$

P1.7 Expand the brackets in the following expressions:

(1) $(x+a)(x-b)$ (2) $(2x+3)(x-5)$ (3) $(5x-2)(2x+7)$

P1.8 Factor the following expressions as a product of linear terms:

(1) $x^2 - 2x - 8$ (2) $3x^3 - 27x$ (3) $6x^2 + 11x - 21$

P1.9 Complete the square in the following quadratic expressions to obtain expressions of the form $A(x-h)^2 + k$.

(1) $x^2 - 4x + 7$ (2) $2x^2 + 12x + 22$ (3) $6x^2 + 11x - 21$

P1.10 A golf club and a golf ball cost \$1.10 together. The golf club costs one dollar more than the ball. How much does the ball cost?

P1.11 An ancient artist drew scenes of hunting on the walls of a cave, including 43 figures of animals and people. There were 17 more figures of animals than people. How many figures of people did the artist draw and how many figures of animals?

P1.12 A father is 35 years old and his son is 5 years old. In how many years will the father's age be four times the son's age?

P1.13 A boy and a girl collected 120 nuts. The girl collected twice as many nuts as the boy. How many nuts did each collect?

P1.14 Alice is 5 years older than Bob. The sum of their ages is 25 years. How old is Alice?

P1.15 A publisher needs to bind 4500 books. One print shop can bind these books in 30 days, another shop can do it in 45 days. How many days are necessary to bind all the books if both shops work in parallel?

Hint: Find the books-per-day rate of each shop.

P1.16 A plane leaves Vancouver travelling at 600 km/h toward Montreal. One hour later, a second plane leaves Vancouver heading for Montreal at 900 km/h. How long will it take for the second plane to overtake the first?

Hint: Distance travelled is equal to velocity multiplied by time: $d = vt$.

P1.17 There are 26 sheep and 10 goats on a ship. How old is the captain?

P1.18 The golden ratio, denoted φ, is the positive solution to the quadratic equation $x^2 - x - 1 = 0$. Find the golden ratio.

P1.19 Solve for x in the equation $\dfrac{1}{x} + \dfrac{2}{1-x} = \dfrac{4}{x^2}$.

Hint: Multiply both sides of the equation by $x^2(1-x)$.

P1.20 Use substitution to solve for x in the following equations:

(1) $x^6 - 4x^3 + 4 = 0$ (2) $\dfrac{1}{2 - \sin x} = \sin x$

P1.21 Find the range of values of the parameter m for which the equation $2x^2 - mx + m = 0$ has no real solutions.

Hint: Use the quadratic formula.

P1.22 Use the properties of exponents and logarithms to simplify

(1) $e^x e^{-x} e^z$ (2) $\left(\dfrac{xy^{-2}z^{-3}}{x^2y^3z^{-4}}\right)^{-3}$ (3) $(8x^6)^{-\frac{2}{3}}$

(4) $\log_4(\sqrt{2})$ (5) $\log_{10}(0.001)$ (6) $\ln(x^2-1) - \ln(x-1)$

P1.23 When representing numbers on a computer, the number of digits of precision n in base b and the approximation error ϵ are related by the equation $n = -\log_b(\epsilon)$. A `float64` has 53 bits of precision (digits base 2). What is the approximation error ϵ for a `float64`? How many digits of precision does a `float64` have in decimal (base 10)?

P1.24 Find the values of x that satisfy the following inequalities:

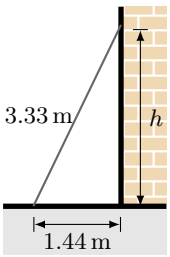
(1) $2x - 5 > 3$ (2) $5 \leq 3x - 4 \leq 14$ (3) $2x^2 + x \geq 1$

P1.25 Two algorithms, P and Q, can be used to solve a certain problem. The running time of Algorithm P as a function of the size of the problem n is described by the function $P(n) = 0.002n^2$. The running time of Algorithm Q is described by $Q(n) = 0.5n$. For small problems, Algorithm P runs faster. Starting from what n will Algorithm Q be faster?

P1.26 Consider a right-angle triangle in which the shorter sides are 8 cm and 6 cm. What is the length of the triangle's longest side?

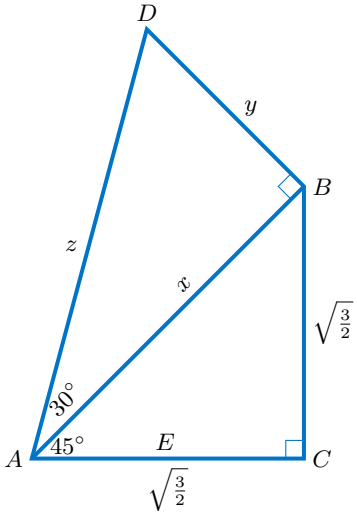
P1.27 A television screen measures 26 inches on the diagonal. The screen height is 13 inches. How wide is the screen?

P1.28 A ladder of length 3.33 m leans against a wall and its foot is 1.44 m from the wall. What is the height h where the ladder touches the wall?



P1.29 **Kepler's triangle** Consider a right-angle triangle in which the hypotenuse has length $\varphi = \frac{\sqrt{5}+1}{2}$ (the golden ratio) and the adjacent side has length $\sqrt{\varphi}$. What is the length of the opposite side?

P1.30 Find the lengths x, y, and z in the figure below.



P1.31 Given the angle and distance measurements labelled in Figure 1.24, calculate the distance d and the height of the mountain peak h.

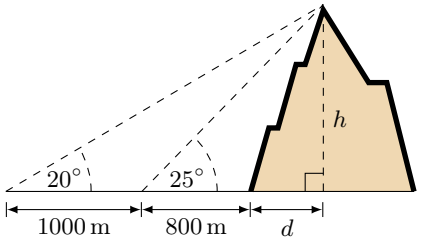
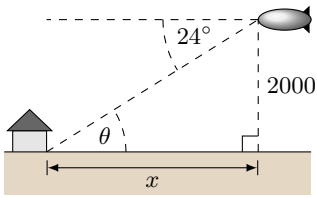


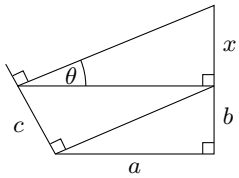
Figure 1.24: Measuring the height of a mountain using angles.

Hint: Use the definition of $\tan\theta$ to obtain two equations in two unknowns.

P1.32 You're observing a house from a blimp flying at an altitude of 2000 metres. From your point of view, the house appears at an angle 24° below the horizontal. What is the horizontal distance x between the blimp and the house?

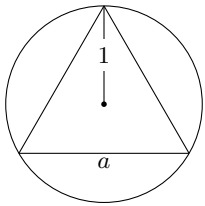


P1.33 Find x. Express your answer in terms of a, b, c and θ.



Hint: Use Pythagoras' theorem twice and the tan function.

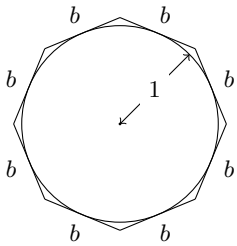
P1.34 An equilateral triangle is inscribed in a circle of radius 1. Find the side length a and the area of the inscribed triangle $A_{\triangle}$.



Hint: Split the triangle into three sub-triangles.

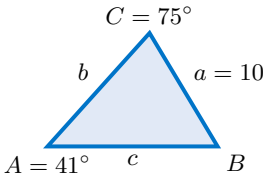
P1.35 Use the power-reduction trigonometric identities (page 64) to express $\sin^2\theta\cos^2\theta$ in terms of $\cos 4\theta$.

P1.36 A circle of radius 1 is inscribed inside a *regular octagon* (a polygon with eight sides of length b). Calculate the octagon's perimeter and its area.



Hint: Split the octagon into eight isosceles triangles.

P1.37 Find the length of side c in the triangle:



Hint: Use the sine rule.

P1.38 Consider the obtuse triangle shown in Figure 1.25.

(a) Express h in terms of a and θ.

(b) What is the area of this triangle?

(c) Express c in terms of the variables a, b, and θ.

Hint: You can use the cosine rule for part (c).

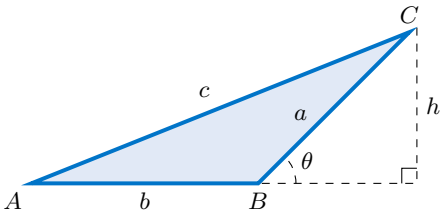
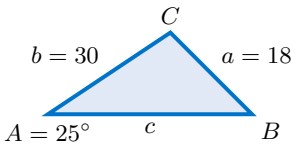


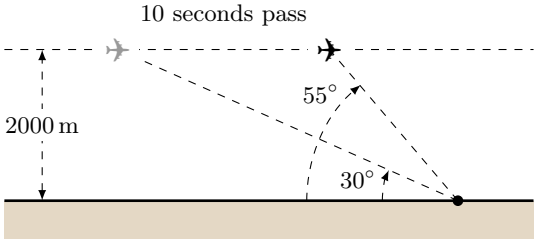
Figure 1.25: A triangle with base b and height h.

P1.39 Find the measure of the angle B and deduce the measure of the angle C. Find the length of side c.



Hint: The sum of the internal angle measures of a triangle is 180°.

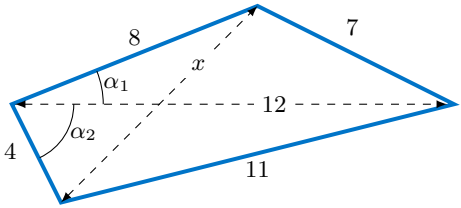
P1.40 An observer on the ground measures an angle of inclination of 30° to an approaching airplane, and 10 seconds later measures an angle of inclination of 55°. If the airplane is flying at a constant speed at an altitude of 2000 m in a straight line directly over the observer, find the speed of the airplane in kilometres per hour.



P1.41 Satoshi likes warm saké. He places 1 litre of water in a sauce pan with diameter 17 cm. How much will the height of the water level rise when Satoshi immerses a saké bottle with diameter 7.5 cm?

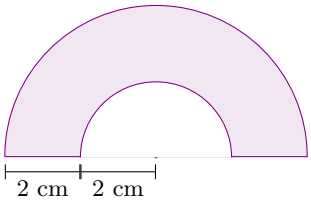
Hint: You'll need the volume conversion ratio 1 litre = $1000\,\text{cm}^3$.

P1.42 Find the length x of the diagonal of the quadrilateral below.



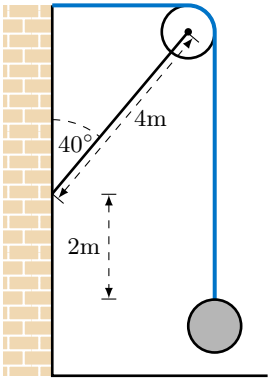
Hint: Use the law of cosines once to find α_1 and α_2, and again to find x.

P1.43 Find the area of the shaded region.



Hint: Find the area of the outer circle, subtract the area of missing centre disk, then divide by two.

P1.44 In preparation for the shooting of a music video, you're asked to suspend a wrecking ball hanging from a circular pulley. The pulley has a radius of 50 cm. The other lengths are indicated in the figure. What is the total length of the rope required?



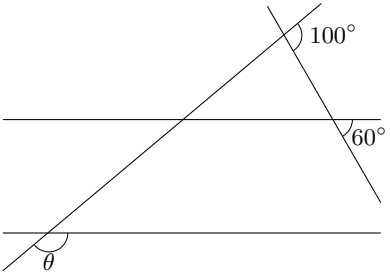
Hint: The total length of rope consists of two straight parts and the curved section that wraps around the pulley.

P1.45 The length of a rectangle is $c + 2$ and its height is 5. What is the area of the rectangle?

P1.46 A box of facial tissues has dimensions 10.5 cm by 7 cm by 22.3 cm. What is the volume of the box in litres?

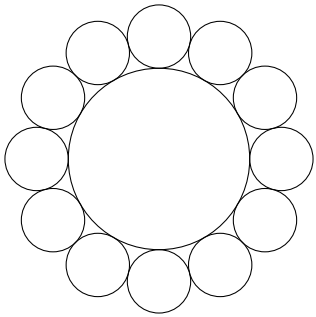
Hint: $1\,\text{L} = 1000\,\text{cm}^3$.

P1.47 What is the measure of the angle θ in the figure below?



Hint: At the intersection of two lines, vertically opposite angles are equal.

P1.48 A large circle of radius R is surrounded by 12 smaller circles of radius r. Find the ratio $\frac{R}{r}$ rounded to four decimals.



Hint: Draw an isosceles triangle with one vertex at the centre of the R-circle and the other vertices at the centres of two adjacent r-circles.

P1.49 The area of a rectangular figure is $35\,\text{cm}^2$. If one side is 5 cm, how long is the other side?

P1.50 A swimming pool has length $\ell = 20\,\text{m}$, width $w = 10\,\text{m}$, and depth $d = 1.5\,\text{m}$. Calculate the volume of water in the swimming pool in litres?

Hint: $1\,\text{m}^3 = 1000\,\text{L}$.

P1.51 How many litres of water remain in a tank that is 12 m long, 6 m wide, and 3 m high, if 30% of its capacity is spent?

P1.52 A building has two water tanks, each with capacity 4000 L. One of them is $\frac{1}{4}$ full and the other contains three times more water. How many litres of water does the building have in total?

P1.53 The rectangular lid of a box has length 40 cm and width 30 cm. A rectangular hole with area $500\,\text{cm}^2$ must be cut in the lid so that the hole's sides are equal distances from the sides of the lid. What will the distance be between the sides of the hole and the sides of the lid?

Hint: You'll need to define three variables to solve this problem.

P1.54 A man sells firewood. To make standard portions, he uses a standard length of rope ℓ to surround a pack of logs. One day, a customer asks him for a double portion of firewood. What length of rope should he use to measure this order? Assume the packs of logs are circular in shape.

P1.55 How much pure water should be added to 10 litres of a solution that is 60% acid to make a solution that is 20% acid?

P1.56 A tablet screen has a resolution of 768 pixels by 1024 pixels, and the physical dimensions of the screen are 6 inches by 8 inches. One might conclude that the best size of a PDF document for such a screen would be 6 inches by 8 inches. At first I thought so too, but I forgot to account for the status bar, which is 20 pixels tall. The actual usable screen area is only 768 pixels by 1004 pixels. Assuming the width of the PDF is chosen to be 6 inches, what height should the PDF be so that it fits perfectly in the content area of the tablet screen?

P1.57 Find the sum of the natural numbers 1 through 100.

Hint: Imagine pairing the biggest number with the smallest number in the sum, the second biggest number with the second smallest number, etc.

P1.58 Solve for x and y simultaneously in the following system of equations: $-x - 2y = -2$ and $3x + 3y = 0$.

P1.59 Solve the following system of equations for the three unknowns:

$$\begin{aligned} 1x + 2y + 3z &= 14, \\ 2x + 5y + 6z &= 30, \\ -1x + 2y + 3z &= 12. \end{aligned}$$

P1.60 A hotel offers a 15% discount on rooms. Determine the original price of a room if the discounted room price is \$95.20.

P1.61 A set of kitchen tools normally retails for \$450, but today it is priced at the special offer of \$360. Calculate the percentage of the discount.

P1.62 You take out a \$5000 loan at a nominal annual percentage rate (nAPR) of 12% with monthly compounding. How much money will you owe after 10 years?

P1.63 Plot the graphs of $f(x) = 100e^{-x/2}$ and $g(x) = 100(1 - e^{-x/2})$ by evaluating the functions at different values of x from 0 to 11.

P1.64 Starting from an initial quantity Q_o of Exponentium at $t = 0\,\text{s}$, the quantity Q of Exponentium as a function of time varies according to the expression $Q(t) = Q_\text{o}\, e^{-\lambda t}$, where $\lambda = 5.0$ and t is measured in seconds. Find the *half-life* of Exponentium, that is, the time it takes for the quantity of Exponentium to reduce to half the initial quantity Q_o.

P1.65 A hot body cools so that every 24 min its temperature decreases by a factor of two. Deduce the time-constant and determine the time it will take the body to reach 1% of its original temperature.

Hint: The temperature function is $T(t) = T_\text{o} e^{-t/\tau}$ and τ is the *time constant*.

P1.66 A capacitor of capacitance $C = 4.0 \times 10^{-6}$ farads, charged to an initial potential of $V_\text{o} = 20\,\text{volts}$, is discharging through a resistance of $R = 10\,000\,\Omega$ (read Ohms). Find the potential V after 0.01 s and after 0.1 s, knowing the decreasing potential follows the rule $V(t) = V_\text{o} e^{-\frac{t}{RC}}$.

P1.67 Let B be the set of people who are bankers and C be the set of crooks. Rewrite the math statement $\exists b \in B \mid b \notin C$ in plain English.

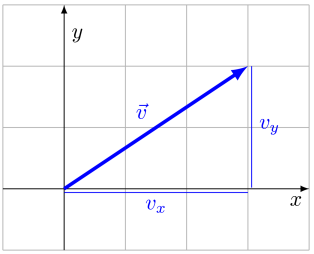
P1.68 Let M denote the set of people who run Monsanto, and H denote the people who ought to burn in hell for all eternity. Write the math statement $\forall p \in M, p \in H$ in plain English.

P1.69 When starting a business, one sometimes needs to find investors. Define M to be the set of investors with money, and C to be the set of investors with connections. Describe the following sets in words: (a) $M \setminus C$, (b) $C \setminus M$, and the most desirable set (c) $M \cap C$.

Chapter 2

Vectors

In this chapter we'll learn how to manipulate multi-dimensional objects called vectors. Vectors are the precise way to describe directions in space. We need vectors in order to describe physical quantities like the velocity of an object, its acceleration, and the net force acting on the object.



Vectors are built from ordinary numbers, which form the *components* of the vector. You can think of a vector as a list of numbers, and *vector algebra* as operations performed on the numbers in the list. Vectors can also be manipulated as geometrical objects, represented by arrows in space. The arrow that corresponds to the vector $\vec{v} = (v_x, v_y)$ starts at the origin $(0, 0)$ and ends at the point (v_x, v_y). The word vector comes from the Latin *vehere*, which means *to carry*. Indeed, the vector $\vec{v}$ takes the point $(0, 0)$ and carries it to the point (v_x, v_y).

This chapter will introduce you to vectors, vector algebra, and vector operations, which are useful for solving problems in many areas of science. What you'll learn here applies more broadly to problems in computer graphics, probability theory, machine learning, and other fields of science and mathematics. It's all about vectors these days, so you better get to know them.

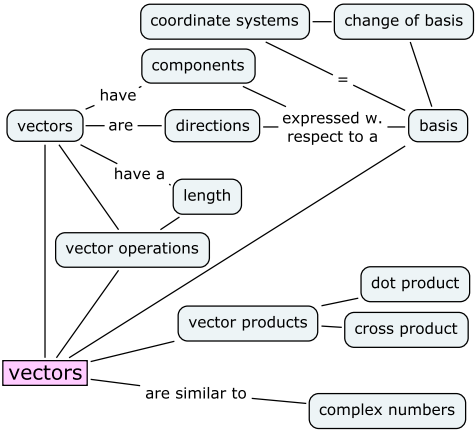


Figure 2.1: This figure illustrates the new concepts related to vectors. As you can see, there is quite a bit of new vocabulary to learn, but don't be phased—all these terms are just fancy ways of talking about arrows.

2.1 Vectors

Vectors are extremely useful in all areas of life. In physics, for example, we use a vector to describe the velocity of an object. It is not sufficient to say that the speed of a tennis ball is 20[m/s]: we must also specify the direction in which the ball is moving. Both of the two velocities

$$\vec{v}_1 = (20, 0) \qquad \text{and} \qquad \vec{v}_2 = (0, 20)$$

describe motion at the speed of 20[m/s]; but since one velocity points along the x-axis, and the other points along the y-axis, they are *completely* different velocities. The velocity vector contains information about the object's speed *and* direction. The direction makes a big difference. If it turns out that the tennis ball is coming your way, you need to get out of the way!

This section's main idea is that **vectors are not the same as numbers**. A vector is a special kind of mathematical object that is *made up of* numbers. Before we begin any calculations with vectors, we need to think about the basic mathematical operations that we can perform on vectors. We will define vector addition $\vec{u} + \vec{v}$, vector subtraction $\vec{u} - \vec{v}$, vector scaling $\alpha\vec{v}$, and other operations. We will also discuss two different notions of *vector product*, which have useful geometrical properties.

Definitions

The two-dimensional vector $\vec{v} \in \mathbb{R}^2$ is equivalent to a *pair of numbers* $\vec{v} \equiv (v_x, v_y)$. We call v_x the x-component of $\vec{v}$, and v_y is the y-component of $\vec{v}$.

Vector representations

We'll use three equivalent ways to denote vectors:

- $\vec{v} = (v_x, v_y)$: component notation, where the vector is represented as a pair of coordinates with respect to the x-axis and the y-axis.
- $\vec{v} = v_x\hat{\imath} + v_y\hat{\jmath}$: unit vector notation, where the vector is expressed in terms of the unit vectors $\hat{\imath} = (1, 0)$ and $\hat{\jmath} = (0, 1)$
- $\vec{v} = \|\vec{v}\|\angle\theta$: length-and-direction notation, where the vector is expressed in terms of its *length* $\|\vec{v}\|$ and the angle θ that the vector makes with the x-axis.

These three notations describe different aspects of vectors, and we will use them throughout the rest of the book. We'll learn how to convert between them—both algebraically (with pen, paper, and calculator) and intuitively (by drawing arrows).

Vector operations

Consider two vectors, $\vec{u} = (u_x, u_y)$ and $\vec{v} = (v_x, v_y)$, and assume that $\alpha \in \mathbb{R}$ is an arbitrary constant. The following operations are defined for these vectors:

- **Addition:** $\vec{u} + \vec{v} = (u_x + v_x, u_y + v_y)$
- **Subtraction:** $\vec{u} - \vec{v} = (u_x - v_x, u_y - v_y)$
- **Scaling:** $\alpha\vec{u} = (\alpha u_x, \alpha u_y)$
- **Dot product:** $\vec{u} \cdot \vec{v} = u_x v_x + u_y v_y$
- **Length:** $\|\vec{u}\| = \sqrt{\vec{u} \cdot \vec{u}} = \sqrt{u_x^2 + u_y^2}$. We will also sometimes simply use the letter u to denote the length of $\vec{u}$.
- **Cross product:** $\vec{u} \times \vec{v} = (u_y v_z - u_z v_y,\ u_z v_x - u_x v_z,\ u_x v_y - u_y v_x)$. The cross product is only defined for three-dimensional vectors like $\vec{u} = (u_x, u_y, u_z)$ and $\vec{v} = (v_x, v_y, v_z)$.

Pay careful attention to the dot product and the cross product. Although they're called products, these operations behave much differently than taking the product of two numbers. Also note, there is no notion of vector division.

Vector algebra

Addition and subtraction Just like numbers, you can add vectors

$$\vec{v} + \vec{w} = (v_x, v_y) + (w_x, w_y) = (v_x + w_x, v_y + w_y),$$

subtract them

$$\vec{v} - \vec{w} = (v_x, v_y) - (w_x, w_y) = (v_x - w_x, v_y - w_y),$$

and solve all kinds of equations where the unknown variable is a vector. This is not a formidably complicated new development in mathematics. Performing arithmetic calculations on vectors simply requires **carrying out arithmetic operations on their components**. Given two vectors, $\vec{v} = (4, 2)$ and $\vec{w} = (3, 7)$, their difference is computed as $\vec{v} - \vec{w} = (4, 2) - (3, 7) = (1, -5)$.

Scaling We can also *scale* a vector by any number $\alpha \in \mathbb{R}$:

$$\alpha\vec{v} = (\alpha v_x, \alpha v_y),$$

where each component is multiplied by the scaling factor α. Scaling changes the length of a vector. If $\alpha > 1$ the vector will get longer, and if $0 \leq \alpha < 1$ then the vector will become shorter. If α is a negative number, the scaled vector will point in the opposite direction.

Length A vector's length is obtained from Pythagoras' theorem. Imagine a right-angle triangle with one side of length v_x and the other side of length v_y; the length of the vector is equal to the length of the triangle's hypotenuse:

$$\|\vec{v}\|^2 = v_x^2 + v_y^2 \qquad \Rightarrow \qquad \|\vec{v}\| = \sqrt{v_x^2 + v_y^2}\,.$$

A common technique is to scale a vector $\vec{v}$ by one over its length $\frac{1}{\|\vec{v}\|}$ to obtain a unit-length vector that points in the same direction as $\vec{v}$:

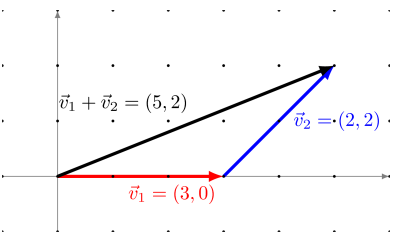
$$\hat{v} \equiv \frac{\vec{v}}{\|\vec{v}\|} = \left(\frac{v_x}{\|\vec{v}\|}, \frac{v_y}{\|\vec{v}\|} \right).$$

Unit vectors (denoted with a hat instead of an arrow) are useful when you want to describe only a direction in space without any specific length in mind. Verify that $\|\hat{v}\| = 1$.

Vector as arrows

So far, we described how to perform algebraic operations on vectors in terms of their components. Vector operations can also be interpreted geometrically, as operations on two-dimensional arrows in the Cartesian plane.

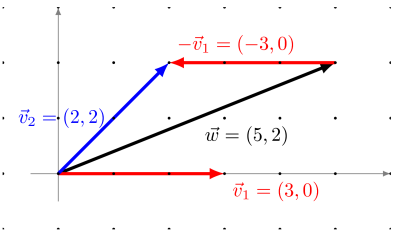
Vector addition The sum of two vectors corresponds to the combined displacement of the two vectors. The diagram on the right illustrates the addition of two vectors, $\vec{v}_1 = (3, 0)$ and $\vec{v}_2 = (2, 2)$. The sum of the two vectors is the vector $\vec{v}_1 + \vec{v}_2 = (3, 0) + (2, 2) = (5, 2)$.



Vector subtraction Before we describe vector subtraction, note that multiplying a vector by a scaling factor $\alpha = -1$ gives a vector of the same length as the original, but pointing in the opposite direction.

This fact is useful if you want to subtract two vectors using the graphical approach. Subtracting a vector is the same as adding the negative of the vector:

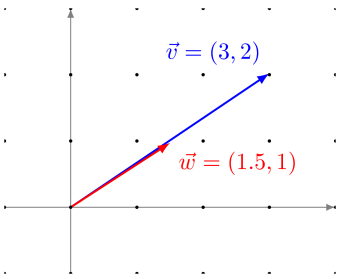
$$\vec{w} - \vec{v}_1 = \vec{w} + (-\vec{v}_1) = \vec{v}_2.$$



The diagram on the right illustrates the graphical procedure for subtracting the vector $\vec{v}_1 = (3, 0)$ from the vector $\vec{w} = (5, 2)$. Subtraction of $\vec{v}_1 = (3, 0)$ is the same as addition of $-\vec{v}_1 = (-3, 0)$.

Scaling The scaling operation acts to change the length of a vector. Suppose we want to obtain a vector in the same direction as the vector $\vec{v} = (3, 2)$, but half as long. "Half as long" corresponds to a scaling factor of $\alpha = 0.5$. The scaled-down vector is $\vec{w} = 0.5\vec{v} = (1.5, 1)$.

Conversely, we can think of the vector $\vec{v}$ as being twice as long as the vector $\vec{w}$.



Length-and-direction representation

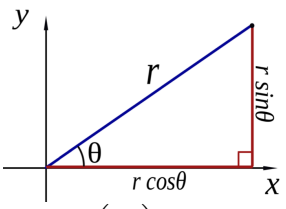
So far, we've seen how to represent a vector in terms of its components. There is also another way of representing vectors: we can specify a vector in terms of its length $||\vec{v}||$ and its direction—the angle it makes with the x-axis. For example, the vector $(1, 1)$ can also be written as

$\sqrt{2}\angle 45°$. This magnitude-and-direction notation is useful because it makes it easy to see the "size" of vectors. On the other hand, vector arithmetic operations are much easier to carry out in the component notation. We will use the following formulas for converting between the two notations.

To convert the length-and-direction vector $\|\vec{r}\|\angle\theta$ into an x-component and a y-component (r_x, r_y), use the formulas

$$r_x = \|\vec{r}\|\cos\theta \quad \text{and} \quad r_y = \|\vec{r}\|\sin\theta.$$

To convert from component notation (r_x, r_y) to length-and-direction $\|\vec{r}\|\angle\theta$, use



$$r = \|\vec{r}\| = \sqrt{r_x^2 + r_y^2} \qquad \text{and} \qquad \theta = \tan^{-1}\left(\frac{r_y}{r_x}\right).$$

Note that the second part of the equation involves the inverse tangent function. By convention, the function $\tan^{-1}$ returns values between $\pi/2$ (90°) and $-\pi/2$ (−90°). You must be careful when finding the θ of vectors with an angle outside of this range. Specifically, for vectors with $v_x < 0$, you must add π (180°) to $\tan^{-1}(r_y/r_x)$ to obtain the correct θ.

Unit vector notation

As discussed above, we can think of a vector $\vec{v} = (v_x, v_y, v_z)$ as a command to "go a distance v_x in the x-direction, a distance v_y in the y-direction, and v_z in the z-direction."

To write this set of commands more explicitly, we can use multiples of the vectors $\hat{\imath}, \hat{\jmath}$, and $\hat{k}$. These are the unit vectors pointing in the x, y, and z directions, respectively:

$$\hat{\imath} = (1,0,0), \qquad \hat{\jmath} = (0,1,0), \quad \text{and} \quad \hat{k} = (0,0,1).$$

Any number multiplied by $\hat{\imath}$ corresponds to a vector with that number in the first coordinate. For example, $3\hat{\imath} \equiv (3,0,0)$. Similarly, $4\hat{\jmath} \equiv (0,4,0)$ and $5\hat{k} \equiv (0,0,5)$.

In physics, we tend to perform a lot of numerical calculations with vectors; to make things easier, we often use unit vector notation:

$$v_x\hat{\imath} + v_y\hat{\jmath} + v_z\hat{k} \qquad \Leftrightarrow \qquad \vec{v} \qquad \Leftrightarrow \qquad (v_x, v_y, v_z).$$

The addition rule remains the same for the new notation:

$$\underbrace{2\hat{\imath} + 3\hat{\jmath}}_{\vec{v}} \; + \; \underbrace{5\hat{\imath} - 2\hat{\jmath}}_{\vec{w}} \; = \; \underbrace{7\hat{\imath} + 1\hat{\jmath}}_{\vec{v}+\vec{w}}.$$

It's the same story repeating all over again: we need to add $\hat{\imath}$s with $\hat{\imath}$s, and $\hat{\jmath}$s with $\hat{\jmath}$s.

Examples

Simple example

Compute the sum $\vec{s} = 4\hat{\imath} + 5\angle 30°$. Express your answer in the length-and-direction notation.

Since we want to carry out an addition, and since addition is performed in terms of components, our first step is to convert $5\angle 30°$ into component notation: $5\angle 30° = 5\cos 30°\hat{\imath} + 5\sin 30°\hat{\jmath} = 5\frac{\sqrt{3}}{2}\hat{\imath} + \frac{5}{2}\hat{\jmath}$. We can now compute the sum:

$$\vec{s} \;=\; 4\hat{\imath} \;+\; 5\tfrac{\sqrt{3}}{2}\hat{\imath} + \tfrac{5}{2}\hat{\jmath} \;=\; (4 + 5\tfrac{\sqrt{3}}{2})\hat{\imath} + (\tfrac{5}{2})\hat{\jmath}\,.$$

The x-component of the sum is $s_x = (4 + 5\frac{\sqrt{3}}{2})$ and the y-component of the sum is $s_y = (\frac{5}{2})$. To express the answer as a length and a direction, we compute the length $\|\vec{s}\| = \sqrt{s_x^2 + s_y^2} = 8.697$ and the direction $\tan^{-1}(s_y/s_x) = 16.7°$. The answer is $\vec{s} = 8.697\angle 16.7°$.

Vector addition example

You're heading to physics class after a "safety meeting" with a friend, and are looking forward to two hours of finding absolute amazement and awe in the laws of Mother Nature. As it turns out, there is no enlightenment to be had that day because there is going to be an in-class midterm. The first question involves a block sliding down an incline. You look at it, draw a little diagram, and then wonder how the hell you are going to find the net force acting on the block. The three forces acting on the block are $\vec{W} = 300\angle -90°$, $\vec{N} = 260\angle 120°$, and $\vec{F}_f = 50\angle 30°$.

You happen to remember the net force formula:

$$\sum \vec{F} = \vec{F}_{\text{net}} = m\vec{a} \qquad [\text{ Newton's } 2^{\text{nd}} \text{ law }].$$

You get the feeling Newton's 2^{nd} law is the answer to all your troubles. You sense this formula is certainly the key because you saw the keyword "net force" when reading the question, and notice "net force" also appears in this very equation.

The net force is the sum of all forces acting on the block:

$$\vec{F}_{\text{net}} = \sum \vec{F} = \vec{W} + \vec{N} + \vec{F}_f.$$

All that separates you from the answer is the addition of these vectors. Vectors have components, and there is the whole sin/cos procedure for decomposing length-and-direction vectors into their components. If you have the vectors as components you'll be able to add them and find the net force.

Okay, chill! Let's do this one step at a time. The net force must have an x-component, which, according to the equation, must equal the sum of the x-components of all the forces:

$$\begin{aligned} F_{\text{net},x} &= W_x + N_x + F_{f,x} \\ &= 300\cos(-90^\circ) + 260\cos(120^\circ) + 50\cos(30^\circ) \\ &= -86.7. \end{aligned}$$

Now find the y-component of the net force using the sin of the angles:

$$\begin{aligned} F_{\text{net},y} &= W_y + N_y + F_{f,y} \\ &= 300\sin(-90^\circ) + 260\sin(120^\circ) + 50\sin(30^\circ) \\ &= -49.8. \end{aligned}$$

Combining the two components of the vector, we get the final answer:

$$\begin{aligned} \vec{F}_{\text{net}} &\equiv (F_{\text{net},x},\, F_{\text{net},y}) \\ &= (-86.7, -49.8) = -86.7\hat{\imath}\ - 49.8\hat{\jmath} \\ &= 100\angle 209.9^\circ. \end{aligned}$$

Bam! Just like that you're done, because you overstand them vectors!

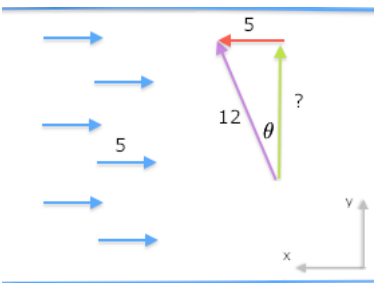
Relative motion example

A boat can reach a top speed of 12 knots in calm seas. Instead of cruising through a calm sea, however, the boat's crew is trying to sail up the St-Laurence river. The speed of the current is 5 knots.

If the boat travels directly upstream at full throttle $12\hat{\imath}$, then the speed of the boat relative to the shore will be

$$12\hat{\imath} - 5\hat{\imath} = 7\hat{\imath},$$

since we have to "deduct" the speed of the current from the speed of the boat relative to the water.



If the crew wants to cross the river perpendicular to the current flow, they can use some of the boat's thrust to counterbalance the current, and the remaining thrust to push across. In what direction should the boat sail to cross the river? We are looking for the direction of $\vec{v}$ the boat should take such that, after adding in the velocity of the current, the boat moves in a straight line between the two banks (the $\hat{\jmath}$ direction).

A picture is necessary: draw a river, then draw a triangle in the river with its long leg perpendicular to the current flow. Make the short leg of length 5. We will take the up-the-river component of the speed $\vec{v}$ to be equal to $5\hat{\imath}$, so that it cancels exactly the $-5\hat{\imath}$ flow of the river. Finally, label the hypotenuse with length 12, since this is the speed of the boat relative to the surface of the water.

From all of this we can answer the question like professionals. You want the angle? Well, we have that $12\sin(\theta) = 5$, where θ is the angle of the boat's course relative to the straight line between the two banks. We can use the inverse-sin function to solve for the angle:

$$\theta = \sin^{-1}\left(\frac{5}{12}\right) = 24.62^\circ.$$

The across-the-river component of the velocity can be calculated using $v_y = 12\cos(\theta) = 10.91$, or from Pythagoras' theorem if you prefer $v_y = \sqrt{\|\vec{v}\|^2 - v_x^2} = \sqrt{12^2 - 5^2} = 10.91$.

Vector dimensions

The most common types of vectors are two-dimensional vectors (like the ones in the Cartesian plane), and three-dimensional vectors (directions in 3D space). 2D and 3D vectors are easier to work with because we can visualize them and draw them in diagrams. In general, vectors can exist in any number of dimensions. An example of a n-dimensional vector is

$$\vec{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^n.$$

The rules of vector algebra apply in higher dimensions, but our ability to visualize stops at three dimensions.

Coordinate system

The geometrical interpretation of vectors depends on the coordinate system in which the vectors are represented. Throughout this section we have used the x, y, and z axes, and we've described vectors as components along each of these directions. This is a very convenient coordinate system; we have a set of three *perpendicular* axes, and a set of three unit vectors $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ that point along each of the three axis directions. Every vector is implicitly defined in terms of this coordinate system. When you and I talk about the vector $\vec{v} = 3\hat{\imath} + 4\hat{\jmath} + 2\hat{k}$, we are really saying, "start from the origin $(0, 0, 0)$, move 3 units in the x-direction, then move 4 units in the y-direction, and finally move 2 units in the z-direction." It is simpler to express these

directions as $\vec{v} = (3, 4, 2)$, while remembering that the numbers in the bracket measure distances *relative* to the xyz-coordinate system.

It turns out, using the xyz-coordinate system and the vectors $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ is just one of many possible ways we can represent vectors. We can represent a vector $\vec{v}$ as coefficients (v_1, v_2, v_3) with respect to any *basis* $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ as follows: $\vec{v} = v_1\hat{e}_1 + v_2\hat{e}_2 + v_3\hat{e}_3$. What is a basis, you ask? I'm glad you asked, because this is the subject of the next section.

2.2 Basis

One of the most important concepts in the study of vectors is the concept of a basis. Consider the space of three-dimensional vectors $\mathbb{R}^3$. A *basis* for $\mathbb{R}^3$ is a set of vectors $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ which can be used as a coordinate system for $\mathbb{R}^3$. If the set of vectors $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ is a basis, then you can *represent* any vector $\vec{v} \in \mathbb{R}^3$ as coefficients (v_1, v_2, v_3) *with respect to* that basis:

$$\vec{v} = v_1\hat{e}_1 + v_2\hat{e}_2 + v_3\hat{e}_3.$$

The vector $\vec{v}$ is obtained by measuring out a distance v_1 in the $\hat{e}_1$ direction, a distance v_2 in the $\hat{e}_2$ direction, and a distance v_3 in the $\hat{e}_3$ direction.

You are already familiar with the *standard* basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$, which is associated with the xyz-coordinate system. You know that any vector $\vec{v} \in \mathbb{R}^3$ can be expressed as a triplet (v_x, v_y, v_z) with respect to the basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ through the formula $\vec{v} = v_x\hat{\imath} + v_y\hat{\jmath} + v_z\hat{k}$. In this section, we'll discuss how to represent vectors with respect to other bases.

An analogy

Let's start with a simple example of a basis. If you look at the HTML code behind any web page, you're sure to find at least one mention of the colour stylesheet directive such as `background-color:#336699;`. The numbers should be interpreted as a triplet of values $(33, 66, 99)$, each value describing the amount of red, green, and blue needed to create a given colour. Let us call the colour described by the triplet $(33, 66, 99)$ CoolBlue. This convention for colour representation is called the RGB colour model and we can think of it as the *RGB basis*. A basis is a set of elements that can be combined together to express something more complicated. In our case, the **R**, **G**, and **B** elements are pure colours that can create any colour when mixed appropriately. Schematically, we can write this mixing idea as

$$\text{CoolBlue} = (33, 66, 99)_{RGB} = 33\mathbf{R} + 66\mathbf{G} + 99\mathbf{B},$$

where the *coefficients* determine the strength of each colour component. To create the colour, we combine its components as symbolized by the + operation.

The cyan, magenta, and yellow (CMY) colour model is another basis for representing colours. To express the "cool blue" colour in the CMY basis, you will need the following coefficients:

$$(33, 66, 99)_{RGB} = \text{CoolBlue} = (222, 189, 156)_{CMY} = 222\mathbf{C} + 189\mathbf{M} + 156\mathbf{Y}.$$

The *same* colour CoolBlue is represented by a *different* set of coefficients when the CMY colour basis is used.

Note that a triplet of coefficients by itself does not mean anything unless we know the basis being used. For example, if we were to interpret the triplet of coordinates $(33, 66, 99)$ with respect to the CMY basis, will would obtain a completely different colour, which would not be cool at all.

A basis is required to convert mathematical objects like the triplet (a, b, c) into real-world ideas like colours. As exemplified above, to avoid any ambiguity we can use a subscript after the bracket to indicate the basis associated with each triplet of coefficients.

Discussion

It's hard to over-emphasize the importance of the basis—the coordinate system you will use to describe vectors. The choice of coordinate system is the bridge between real-world vector quantities and their mathematical representation in terms of components. Every time you solve a problem with vectors, **the first thing you should do is draw a coordinate system**, and think of vector components as measuring out a distance along this coordinate system.

2.3 Vector products

If addition of two vectors $\vec{v}$ and $\vec{w}$ corresponds to the addition of their components $(v_x + w_x, v_y + w_y, v_z + w_z)$, you might logically think that the product of two vectors will correspond to the product of their components $(v_x w_x, v_y w_y, v_z w_z)$, however, this way of multiplying vectors is not used in practice. Instead, we use the dot product and the cross product.

The *dot product* tells you how similar two vectors are to each other:

$$\vec{v} \cdot \vec{w} \equiv v_x w_x + v_y w_y + v_z w_z \equiv \|\vec{v}\|\|\vec{w}\| \cos(\varphi) \quad \in \mathbb{R},$$

where φ is the angle between the two vectors. The factor $\cos(\varphi)$ is largest when the two vectors point in the same direction because the angle between them will be $\varphi = 0$ and $\cos(0) = 1$.

The exact formula for the *cross product* is more complicated so I will not show it to you just yet. What is important to know is that the cross product of two vectors is another vector:

$$\vec{v} \times \vec{w} = \{ \text{ a vector perpendicular to both } \vec{v} \text{ and } \vec{w} \ \} \quad \in \mathbb{R}^3.$$

If you take the cross product of one vector pointing in the x-direction with another vector pointing in the y-direction, the result will be a vector in the z-direction.

Dot product

The *dot product* takes two vectors as inputs and produces a single real number as an output:

$$\cdot : \mathbb{R}^3 \times \mathbb{R}^3 \quad \to \quad \mathbb{R}.$$

The dot product between two vectors can be computed using either the algebraic formula

$$\vec{v} \cdot \vec{w} \equiv v_x w_x + v_y w_y + v_z w_z,$$

or the geometrical formula

$$\vec{v} \cdot \vec{w} \equiv \|\vec{v}\| \|\vec{w}\| \cos(\varphi),$$

where φ is the angle between the two vectors.

The dot product is also known as the *inner* product or *scalar* product. The name *scalar* comes from the fact that the result of the dot product is a scalar number—a number that does not change when the basis changes.

We can combine the algebraic and the geometric formulas for the dot product to obtain the formula

$$\cos(\varphi) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \|\vec{w}\|} = \frac{v_x w_x + v_y w_y + v_z w_z}{\|\vec{v}\| \|\vec{w}\|} \quad \text{and} \quad \varphi = \cos^{-1}(\cos(\varphi)).$$

Thus, it's possible to find the angle between two vectors if we know their components.

The geometric factor $\cos(\varphi)$ depends on the relative orientation of the two vectors as follows:

- If the vectors point in the same direction, then $\cos(\varphi) = \cos(0^\circ) = 1$ and so $\vec{v} \cdot \vec{w} = \|\vec{v}\| \|\vec{w}\|$.
- If the vectors are perpendicular to each other, then $\cos(\varphi) = \cos(90^\circ) = 0$ and so $\vec{v} \cdot \vec{w} = \|\vec{v}\| \|\vec{w}\| (0) = 0$.
- If the vectors point in exactly opposite directions, then $\cos(\varphi) = \cos(180^\circ) = -1$ and so $\vec{v} \cdot \vec{w} = -\|\vec{v}\| \|\vec{w}\|$.

Cross product

The *cross product* takes two vectors as inputs and produces another vector as the output:

$$\times : \mathbb{R}^3 \times \mathbb{R}^3 \quad \to \quad \mathbb{R}^3.$$

Because the output of this operation is a vector, we sometimes refer to the cross product as the *vector* product.

The cross products of individual basis elements are defined as follows:

$$\hat{\imath} \times \hat{\jmath} = \hat{k}, \quad \hat{\jmath} \times \hat{k} = \hat{\imath}, \quad \hat{k} \times \hat{\imath} = \hat{\jmath}.$$

The cross product is *anti-symmetric* in its inputs, which means swapping the order of the inputs introduces a negative sign in the output:

$$\hat{\jmath} \times \hat{\imath} = -\hat{k}, \quad \hat{k} \times \hat{\jmath} = -\hat{\imath}, \quad \hat{\imath} \times \hat{k} = -\hat{\jmath}.$$

I bet you had never seen a product like this before. Most likely, the products you've seen in math have been *commutative*, which means the order of the inputs doesn't matter. The product of two numbers is commutative $ab = ba$, and the dot product is commutative $\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u}$, but the cross product of two vectors is *non-commutative* $\hat{\imath} \times \hat{\jmath} \neq \hat{\jmath} \times \hat{\imath}$.

For two arbitrary vectors $\vec{a} = (a_x, a_y, a_z)$ and $\vec{b} = (b_x, b_y, b_z)$, the cross product is calculated as

$$\vec{a} \times \vec{b} = (a_y b_z - a_z b_y,\ a_z b_x - a_x b_z,\ a_x b_y - a_y b_x)\,.$$

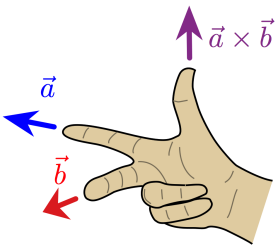
The cross product's output has a length that is proportional to the sin of the angle between the vectors:

$$\|\vec{a} \times \vec{b}\| = \|\vec{a}\| \|\vec{b}\| \sin(\varphi).$$

The direction of the vector $(\vec{a} \times \vec{b})$ is perpendicular to both $\vec{a}$ and $\vec{b}$.

The right-hand rule

Consider the plane formed by the vectors $\vec{a}$ and $\vec{b}$. There are actually *two* vectors that are perpendicular to this plane: one above the plane and one below the plane. We use the *right-hand rule* to figure out which of these vectors corresponds to the cross product $\vec{a} \times \vec{b}$.



When your index finger points in the same direction as the vector $\vec{a}$ and your middle finger points in the direction of $\vec{b}$, your thumb will point in the direction of $\vec{a} \times \vec{b}$. The relationship encoded in the right-hand rule matches the relationship between the standard basis vectors: $\hat{\imath} \times \hat{\jmath} = \hat{k}$.

Links

[A nice illustration of the cross product]
http://1ucasvb.tumblr.com/post/76812811092/

2.4 Complex numbers

By now, you've heard about complex numbers $\mathbb{C}$. The word "complex" is an intimidating word. Surely it must be a complex task to learn about the complex numbers. That may be true in general, but it helps if you know about vectors. Complex numbers are similar to two-dimensional vectors $\vec{v} \in \mathbb{R}^2$. We add and subtract complex numbers like vectors. Complex numbers also have components, length, and "direction." If you understand vectors, you will understand complex numbers at almost no additional mental cost.

We'll begin with a practical problem.

Example

Suppose you're asked to solve the following quadratic equation:

$$x^2 + 1 = 0.$$

You're looking for a number x, such that $x^2 = -1$. If you are only allowed to give real answers (the set of real numbers is denoted $\mathbb{R}$), then there is no answer to this question. In other words, this equation has no solutions. Graphically speaking, this is because the quadratic function $f(x) = x^2 + 1$ does not cross the x-axis.

However, we're not taking no for an answer! If we insist on solving for x in the equation $x^2 + 1 = 0$, we can imagine a new number i that satisfies $i^2 = -1$. We call i the unit imaginary number. The solutions to the equation are therefore $x_1 = i$ and $x_2 = -i$. There are two solutions because the equation was quadratic. We can check that $i^2 + 1 = -1 + 1 = 0$ and also $(-i)^2 + 1 = (-1)^2 i^2 + 1 = i^2 + 1 = 0$.

Thus, while the equation $x^2 + 1 = 0$ has no real solutions, it *does* have solutions if we allow the answers to be imaginary numbers.

Definitions

Complex numbers have a real part and an imaginary part:

- i: the unit imaginary number $i \equiv \sqrt{-1}$ or $i^2 = -1$
- bi: an imaginary number that is equal to b times i
- $\mathbb{R}$: the set of real numbers
- $\mathbb{C}$: the set of complex numbers $\mathbb{C} = \{a + bi \mid a, b \in \mathbb{R}\}$

- $z = a + bi$: a complex number
- $\mathrm{Re}\{z\} = a$: the real part of z
- $\mathrm{Im}\{z\} = b$: the imaginary part of z
- $\bar{z}$: the *complex conjugate* of z. If $z = a + bi$, then $\bar{z} = a - bi$.

The polar representation of complex numbers:

- $z = |z|\angle\varphi_z = |z|\cos\varphi_z + i|z|\sin\varphi_z$
- $|z| = \sqrt{\bar{z}z} = \sqrt{a^2 + b^2}$: the *magnitude* of $z = a + bi$
- $\varphi_z = \tan^{-1}(b/a)$: the *phase* or *argument* of $z = a + bi$
- $\mathrm{Re}\{z\} = |z|\cos\varphi_z$
- $\mathrm{Im}\{z\} = |z|\sin\varphi_z$

Formulas

Addition and subtraction

Just as we performed the addition of vectors component by component, we perform addition on complex numbers by adding the real parts together and adding the imaginary parts together:

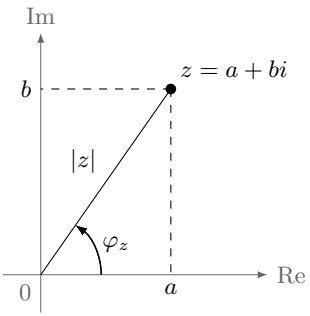
$$(a + bi) + (c + di) = (a + c) + (b + d)i.$$

Polar representation

We can give a geometrical interpretation of the complex numbers by extending the real number line into a two-dimensional plane called the *complex plane*. The horizontal axis in the complex plane measures the *real* part of the number. The vertical axis measures the *imaginary* part. Complex numbers are vectors in the complex plane.

It is possible to represent any complex number $z = a + bi$ in terms of its *magnitude* and its *phase*:

$$z = |z|\angle\varphi_z = \underbrace{|z|\cos\varphi_z}_{a} + \underbrace{|z|\sin\varphi_z}_{b}\, i.$$



The magnitude of a complex number $z = a + bi$ is

$$|z| = \sqrt{a^2 + b^2}.$$

This corresponds to the *length* of the vector that represents the complex number in the complex plane. The formula is obtained by using Pythagoras' theorem.

The *phase*, also known as the *argument* of the complex number $z = a + bi$ is

$$\varphi_z \equiv \arg z \;=\; \texttt{atan2(b,a)} \;=^{\dagger}\; \tan^{-1}(b/a).$$

The phase corresponds to the angle z forms with the real axis. Note the equality labelled † is true only when $a > 0$, because the function $\tan^{-1}$ always returns numbers in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Manual corrections of the output of $\tan^{-1}(b/a)$ are required for complex numbers with $a < 0$.

Some programming languages provide the 2-input math function `atan2(y,x)` that correctly computes the angle the vector (x, y) makes with the x-axis in all four quadrants. Complex numbers behave like 2-dimensional vectors so you can use `atan2` to compute their phase.

Complex numbers have vector-like properties like magnitude and phase, but we can also do other operations with them that are not defined for vectors. The set of complex numbers $\mathbb{C}$ is a *field*. This means, in addition to the addition and subtraction operations, we can also perform multiplication and division with complex numbers.

Multiplication

The product of two complex numbers is computed using the usual rules of algebra:

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i.$$

In the polar representation, the product formula is

$$(p\angle\phi)(q\angle\psi) = pq\angle(\phi + \psi).$$

To multiply two complex numbers, multiply their magnitudes and add their phases.

Division

Let's look at the procedure for dividing complex numbers:

$$\frac{(a + bi)}{(c + di)} = \frac{(a + bi)}{(c + di)}\frac{(c - di)}{(c - di)} = (a + bi)\frac{(c - di)}{(c^2 + d^2)} = (a + bi)\frac{\overline{c + di}}{|c + di|^2}.$$

In other words, to divide the number z by the complex number s, compute $\bar{s}$ and $|s|^2 = s\bar{s}$ and then use

$$z/s = z\frac{\bar{s}}{|s|^2}.$$

You can think of $\frac{\bar{s}}{|s|^2}$ as being equivalent to s^{-1}.

Cardano's example One of the earliest examples of reasoning involving complex numbers was given by Gerolamo Cardano in his 1545 book *Ars Magna*. Cardano wrote, "If someone says to you, divide 10 into two parts, one of which multiplied into the other shall produce 40, it is evident that this case or question is impossible." We want to find numbers x_1 and x_2 such that $x_1 + x_2 = 10$ and $x_1 x_2 = 40$. This sounds kind of impossible. Or is it?

"Nevertheless," Cardano said, "we shall solve it in this fashion:

$$x_1 = 5 + \sqrt{15}i \quad \text{and} \quad x_2 = 5 - \sqrt{15}i."$$

When you add $x_1 + x_2$ you obtain 10. When you multiply the two numbers the answer is

$$\begin{aligned} x_1 x_2 &= \left(5 + \sqrt{15}i\right)\left(5 - \sqrt{15}i\right) \\ &= 25 - 5\sqrt{15}i + 5\sqrt{15}i - \sqrt{15}^2 i^2 = 25 + 15 = 40. \end{aligned}$$

Hence $5 + \sqrt{15}i$ and $5 - \sqrt{15}i$ are two numbers whose sum is 10 and whose product is 40.

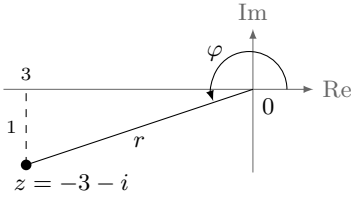
Example 2 Compute the product of i and -1. Both i and -1 have a magnitude of 1 but different phases. The phase of i is $\frac{\pi}{2}$ (90°), while -1 has phase π (180°). The product of these two numbers is

$$(i)(-1) \;=\; (1\angle\tfrac{\pi}{2})(1\angle\pi) \;=\; 1\angle(\tfrac{\pi}{2} + \pi) \;=\; 1\angle\tfrac{3\pi}{2} \;=\; -i.$$

Multiplication by i is effectively a rotation by $\frac{\pi}{2}$ (90°) to the left.

Example 3 Find the polar representation of $z = -3 - i$ and compute z^6.

Let's denote the polar representation of z by $z = r\angle\varphi$. We find $r = \sqrt{3^2 + 1^2} = \sqrt{10}$ and $\varphi = \tan^{-1}(\frac{1}{3}) + \pi = 0.322 + \pi$.



Using the polar representation, we can easily compute z^6:

$$z^6 = r^6\angle(6\varphi) = (\sqrt{10})^6\angle 6(0.322+\pi) = 10^3\angle 1.932 + 6\pi = 10^3\angle 1.932.$$

Note we can ignore multiples of 2π in the phase. In component form, z^6 is equal to $1000\cos(1.932) + 1000\sin(1.932)i = -353.4 + 935.5i$.

Fundamental theorem of algebra

The solutions to *any* polynomial equation $a_0 + a_1 x + \cdots + a_n x^n = 0$ are of the form

$$z = a + bi.$$

In other words, any polynomial $P(x)$ of n^{th} degree can be written as

$$P(x) = (x - z_1)(x - z_2) \cdots (x - z_n),$$

where $z_i \in \mathbb{C}$ are the polynomial's *complex* roots. Before today, you might have said the equation $x^2 + 1 = 0$ has no solutions. Now you know its solutions are the complex numbers $z_1 = i$ and $z_2 = -i$.

The theorem is "fundamental" because it tells us we won't ever need to invent any "fancier" set of numbers to solve polynomial equations. Recall that each set of numbers is associated with a different class of equations. The natural numbers $\mathbb{N}$ appear as solutions of the equation $m + n = x$, where m and n are natural numbers (denoted $m, n \in \mathbb{N}$). The integers $\mathbb{Z}$ are the solutions to equations of the form $x + m = n$, where $m, n \in \mathbb{N}$. The rational numbers $\mathbb{Q}$ are necessary to solve for x in $mx = n$, with $m, n \in \mathbb{Z}$. To find the solutions of $x^2 = 2$, we need the real numbers $\mathbb{R}$. The process of requiring new types of numbers for solving more complicated types of equations stops at $\mathbb{C}$; any polynomial equation—no matter how complicated it is—has solutions that are complex numbers $\mathbb{C}$.

Euler's formula

You already know $\cos\theta$ is a shifted version of $\sin\theta$, so it's clear these two functions are related. It turns out the exponential function is also related to sin and cos. Lo and behold, we have Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta\,.$$

Inputting an imaginary number to the exponential function outputs a complex number that contains both cos and sin. Euler's formula gives us an alternate notation for the polar representation of complex numbers: $z = |z|\angle\varphi_z = |z|e^{i\varphi_z}$.

If you want to impress your friends with your math knowledge, plug $\theta = \pi$ into the above equation to find

$$e^{i\pi} = \cos(\pi) + i\sin(\pi) = -1,$$

which can be rearranged into the form, $e^{\pi i} + 1 = 0$. This equation shows a relationship between the five most important numbers in all of mathematics: Euler's number $e = 2.71828\ldots$, $\pi = 3.14159\ldots$, the imaginary number i, 1, and zero. It's kind of cool to see all these important numbers reunited in one equation, don't you agree?

De Moivre's theorem

By replacing θ in Euler's formula with $n\theta$, we obtain de Moivre's theorem:

$$(\cos\theta + i\sin\theta)^n = \cos n\theta + i\sin n\theta.$$

De Moivre's Theorem makes sense if you think of the complex number $z = e^{i\theta} = \cos\theta + i\sin\theta$, raised to the n^{th} power:

$$(\cos\theta + i\sin\theta)^n = z^n = (e^{i\theta})^n = e^{in\theta} = \cos n\theta + i\sin n\theta.$$

Setting $n = 2$ in de Moivre's formula, we can derive the double angle formulas (page 64) as the real and imaginary parts of the following equation:

$$(\cos^2\theta - \sin^2\theta) + (2\sin\theta\cos\theta)i = \cos(2\theta) + \sin(2\theta)i.$$

Links

[Mini tutorial on the complex numbers]
`http://paste.lisp.org/display/133628`

2.5 Vectors problems

You learned a bunch of vector formulas and you saw some vector diagrams, but did you really learn how to solve problems with vectors? There is only one way to find out: test yourself by solving problems.

I've said it before and I don't want to repeat myself too much, but it's worth saying again: the more problems you solve, the better you'll understand the material. It's now time for you to try the following vector problems to make sure you're on top of things.

P2.1 Express the following vectors in length-and-direction notation:

(a) $\vec{u}_1 = (0, 5)$ (b) $\vec{u}_2 = (1, 2)$ (c) $\vec{u}_3 = (-1, -2)$

P2.2 Express the following vectors as components:

(a) $\vec{v}_1 = 20\angle 30^\circ$ (b) $\vec{v}_2 = 10\angle -90^\circ$ (c) $\vec{v}_3 = 5\angle 150^\circ$

P2.3 Express the following vectors in terms of unit vectors $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$:

(a) $\vec{w}_1 = 10\angle 25^\circ$ (b) $\vec{w}_2 = 7\angle -90^\circ$ (c) $\vec{w}_3 = (3, -2, 3)$

P2.4 Given the vectors $\vec{v}_1 = (1, 1)$, $\vec{v}_2 = (2, 3)$, and $\vec{v}_3 = 5\angle 30^\circ$, calculate the following expressions:

(a) $\vec{v}_1 + \vec{v}_2$ (b) $\vec{v}_2 - 2\vec{v}_1$ (c) $\vec{v}_1 + \vec{v}_2 + \vec{v}_3$

P2.5 Starting from the point $P = (2, 6)$, the three displacement vectors shown in Figure 2.2 are applied to obtain the point Q. What are the coordinates of the point Q?

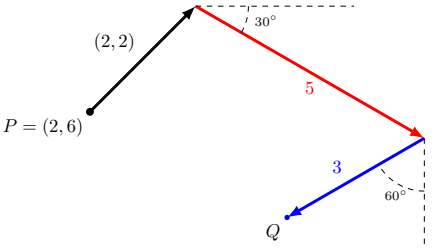


Figure 2.2: A point P is displaced by three vectors to obtain point Q.

P2.6 Given the vectors $\vec{u} = (1, 1, 1)$, $\vec{v} = (2, 3, 1)$, and $\vec{w} = (-1, -1, 2)$, compute the following products:

(1) $\vec{u} \cdot \vec{v}$ (2) $\vec{u} \cdot \vec{w}$ (3) $\vec{v} \cdot \vec{w}$

(4) $\vec{u} \times \vec{v}$ (5) $\vec{u} \times \vec{w}$ (6) $\vec{v} \times \vec{w}$

P2.7 Find a unit-length vector that is perpendicular to both $\vec{u} = (1, 0, 1)$ and $\vec{v} = (1, 2, 0)$.

Hint: Use the cross product.

P2.8 Find a vector that is orthogonal to both $\vec{u}_1 = (1, 0, 1)$ and $\vec{u}_2 = (1, 3, 0)$ and whose dot product with the vector $\vec{v} = (1, 1, 0)$ is equal to 8.

P2.9 Compute the following expressions:

(a) $\sqrt{-4}$ (b) $\dfrac{2 + 3i}{2 + 2i}$ (c) $e^{3i}(2 + i)e^{-3i}$

P2.10 Solve for $x \in \mathbb{C}$ in the following equations:

(a) $x^2 = -4$ (b) $\sqrt{x} = 4i$

(c) $x^2 + 2x + 2 = 0$ (d) $x^4 + 4x^2 + 3 = 0$

Hint: To solve (d), use the substitution $u = x^2$.

P2.11 Given the numbers $z_1 = 2+i$, $z_2 = 2-i$, and $z_3 = -1-i$, compute

(a) $|z_1|$ (b) $\dfrac{z_1}{z_3}$ (c) $z_1 z_2 z_3$

P2.12 A real business is a business that is profitable. An imaginary business is an idea that is just turning around in your head. We can model the real-imaginary nature of a business project by representing the *project state* as a complex number $p \in \mathbb{C}$. For example, a business idea is described by the state $p_o = 100i$. In other words, it is 100% imaginary.

To bring an idea from the imaginary into the real, you must work on it. We'll model the work done on the project as a multiplication by the complex number $e^{-i\alpha h}$, where h is the number of hours of work and α is a constant that depends on the project. After h hours of work, the initial state of the project is transformed as follows: $p_f = e^{-i\alpha h} p_i$. Working on the project for one hour "rotates" its state by $-\alpha$[rad], making it less imaginary and more real.

If you start from an idea $p_o = 100i$ and the cumulative number of hours invested after t weeks of working on the project is $h(t) = 0.2t^2$, how long will it take for the project to become 100% real? Assume $\alpha = 2.904 \times 10^{-3}$.

Hint: A project is 100% real if $\text{Re}\{p\} = p$.

Chapter 3

Intro to linear algebra

The first two chapters reviewed core ideas of mathematics, along with basic notions about vectors you may have been exposed to when learning physics. Now that we're done with the prerequisites, it's time to dig into the main discussion of linear algebra: the study of vectors and matrices.

3.1 Introduction

Vectors and matrices are the new objects of study in linear algebra, and our first step is to define the basic operations we can perform on them.

We denote the set of n-dimensional vectors as $\mathbb{R}^n$. A vector $\vec{v} \in \mathbb{R}^n$ is an n-tuple of real numbers.[1] For example, a three-dimensional vector is a triple of the form

$$\vec{v} = (v_1, v_2, v_3) \quad \in \quad (\mathbb{R}, \mathbb{R}, \mathbb{R}) \equiv \mathbb{R}^3.$$

To specify the vector $\vec{v}$, we must specify the values for its three components: v_1, v_2, and v_3.

A matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular array of real numbers with m rows and n columns. For example, a 3×2 matrix looks like this:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \in \begin{bmatrix} \mathbb{R} & \mathbb{R} \\ \mathbb{R} & \mathbb{R} \\ \mathbb{R} & \mathbb{R} \end{bmatrix} \equiv \mathbb{R}^{3 \times 2}.$$

To specify the matrix A, we need to specify the values of its six components, a_{11}, a_{12}, $\ldots$, a_{32}.

[1] The notation "$s \in S$" is read "s element of S" or "s in S."

In the remainder of this chapter we'll learn about the mathematical operations we can perform on vectors and matrices. Many problems in science, business, and technology can be described in terms of vectors and matrices so it's important you understand how to work with these math objects.

Context

To illustrate what is new about vectors and matrices, let's begin by reviewing the properties of something more familiar: the set of real numbers $\mathbb{R}$. The basic operations on numbers are:

- Addition (denoted $+$)
- Subtraction, the inverse of addition (denoted $-$)
- Multiplication (denoted implicitly)
- Division, the inverse of multiplication (denoted by fractions)

You're familiar with these operations and know how to use them to evaluate math expressions and solve equations.

You should also be familiar with *functions* that take real numbers as inputs and give real numbers as outputs, denoted $f : \mathbb{R} \to \mathbb{R}$. Recall that, by definition, the *inverse function* f^{-1} *undoes* the effect of f. If you are given $f(x)$ and want to find x, you can use the inverse function as follows: $f^{-1}(f(x)) = x$. For example, the function $f(x) = \ln(x)$ has the inverse $f^{-1}(x) = e^x$, and the inverse of $g(x) = \sqrt{x}$ is $g^{-1}(x) = x^2$.

Vector operations

The operations we can perform on vectors are:

- Addition (denoted $+$)
- Subtraction, the inverse of addition (denoted $-$)
- Scaling
- Dot product (denoted $\cdot$)
- Cross product (denoted $\times$)

We'll discuss each of these vector operations in Section 3.2. Although you should already be familiar with vectors and vector operations from Chapter 2, it's worth revisiting and reviewing these concepts because vectors are the foundation of linear algebra.

Matrix operations

The mathematical operations defined for matrices A and B are:

- Addition (denoted $A + B$)
- Subtraction, the inverse of addition (denoted $A - B$)
- Scaling by a constant α (denoted αA)
- Matrix product (denoted AB)
- Matrix inverse (denoted A^{-1})
- Trace (denoted $\text{Tr}(A)$)
- Determinant (denoted $\det(A)$ or $|A|$)

We'll define each of these operations in Section 3.3. We'll learn about the various computational, geometrical, and theoretical considerations associated with these matrix operations in the remainder of the book.

Let's now examine one important matrix operation in closer detail: the matrix-vector product $A\vec{x}$.

Matrix-vector product

Consider the matrix $A \in \mathbb{R}^{m \times n}$ and the vector $\vec{v} \in \mathbb{R}^n$. The matrix-vector product $A\vec{x}$ produces a linear combination of the columns of the matrix A with coefficients $\vec{x}$. For example, the product of a 3×2 matrix A and a 2×1 vector $\vec{x}$ results in a 3×1 vector, which we'll denote $\vec{y}$:

$$\vec{y} = A\vec{x},$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \equiv \begin{bmatrix} x_1 a_{11} + x_2 a_{12} \\ x_1 a_{21} + x_2 a_{22} \\ x_1 a_{31} + x_2 a_{32} \end{bmatrix} \equiv \underbrace{x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}}_{\text{column picture}}.$$

The key thing to observe in the above formula is the interpretation of the matrix-vector product in the *column picture*: $\vec{y} = A\vec{x} = x_1 A_{[:,1]} + x_2 A_{[:,2]}$, where $A_{[:,1]}$ and $A_{[:,2]}$ are the first and second columns of A. For example, if you want to obtain the linear combination consisting of 3 times the first column of A and 4 times the second column of A, you can multiply A by the vector $\vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Linear combinations as matrix products

Consider some set of vectors $\{\vec{e}_1, \vec{e}_2\}$, and a third vector $\vec{y}$ that is a *linear combination* of the vectors $\vec{e}_1$ and $\vec{e}_2$:

$$\vec{y} \;=\; \alpha\vec{e}_1 \;+\; \beta\vec{e}_2.$$

The numbers $\alpha, \beta \in \mathbb{R}$ are the *coefficients* in this linear combination.

The matrix-vector product is defined *expressly* for the purpose of studying linear combinations. We can describe the above linear combination as the following matrix-vector product:

$$\vec{y} = \begin{bmatrix} | & | \\ \vec{e}_1 & \vec{e}_2 \\ | & | \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = E\vec{x}.$$

The matrix E has $\vec{e}_1$ and $\vec{e}_2$ as columns. The dimensions of the matrix E will be $n \times 2$, where n is the dimension of the vectors $\vec{e}_1$, $\vec{e}_2$, and $\vec{y}$.

Vector functions

Okay dear readers, we've reached the key notion in the study of linear algebra. This is the crux. The essential fibre. The meat and potatoes. The *main idea*. I know you're ready to handle it because you're familiar with functions of a real variable $f : \mathbb{R} \to \mathbb{R}$, and you just learned the definition of the matrix-vector product (in which the variables were chosen to subliminally remind you of the standard conventions for the function input x and the function output $y = f(x)$). Without further ado, I present to you the concept of a vector function.

The matrix-vector product corresponds to the abstract notion of a *linear transformation*, which is one of the key notions in the study of linear algebra. Multiplication by a matrix $A \in \mathbb{R}^{m \times n}$ can be thought of as computing a *linear transformation* T_A that takes n-vectors as inputs and produces m-vectors as outputs:

$$T_A : \mathbb{R}^n \to \mathbb{R}^m.$$

Instead of writing $\vec{y} = T_A(\vec{x})$ to denote the linear transformation T_A applied to the vector $\vec{x}$, we can write $\vec{y} = A\vec{x}$. Since the matrix A has m rows, the result of the matrix-vector product is an m-vector. Applying the linear transformation T_A to the vector $\vec{x}$ corresponds to the product of the matrix A and the column vector $\vec{x}$. We say T_A is *represented by* the matrix A.

Inverse When a matrix A is square and invertible, there exists an inverse matrix A^{-1} which *undoes* the effect of A to restore the original input vector:

$$A^{-1}(A(\vec{x})) = A^{-1}A\vec{x} = \vec{x}.$$

Using the matrix inverse A^{-1} to undo the effects of the matrix A is analogous to using the inverse function f^{-1} to undo the effects of the function f.

Example 1 Consider the linear transformation that multiplies the first components of input vectors by 3 and multiplies the second components by 5, as described by the matrix

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix}, \qquad A(\vec{x}) = \begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3x_1 \\ 5x_2 \end{bmatrix}.$$

Its inverse is

$$A^{-1} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{5} \end{bmatrix}, \qquad A^{-1}(A(\vec{x})) = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{5} \end{bmatrix}\begin{bmatrix} 3x_1 \\ 5x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \vec{x}.$$

The inverse matrix multiplies the first component by $\frac{1}{3}$ and the second component by $\frac{1}{5}$, which effectively undoes what A did.

Example 2 Things get a little more complicated when matrices *mix* the different components of the input vector, as in the following example:

$$B = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \quad \text{which acts as} \quad B(\vec{x}) = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + 2x_2 \\ 3x_2 \end{bmatrix}.$$

Make sure you understand how to compute $B(\vec{x}) \equiv B\vec{x}$ using the *column picture* of the matrix-vector product.

The inverse of the matrix B is the matrix

$$B^{-1} = \begin{bmatrix} 1 & \frac{-2}{3} \\ 0 & \frac{1}{3} \end{bmatrix}.$$

Multiplication by the matrix B^{-1} is the "undo action" for multiplication by B:

$$B^{-1}(B(\vec{x})) = \begin{bmatrix} 1 & \frac{-2}{3} \\ 0 & \frac{1}{3} \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & \frac{-2}{3} \\ 0 & \frac{1}{3} \end{bmatrix}\begin{bmatrix} x_1 + 2x_2 \\ 3x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \vec{x}.$$

By definition, the inverse A^{-1} *undoes* the effects of the matrix A. The cumulative effect of applying A^{-1} after A is the *identity matrix* $\mathbb{1}$, which has 1s on the diagonal and 0s everywhere else:

$$A^{-1}A\vec{x} = \mathbb{1}\vec{x} = \vec{x} \qquad \Rightarrow \qquad A^{-1}A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbb{1}.$$

Note that $\mathbb{1}\vec{x} = \vec{x}$ for any vector $\vec{x}$.

We'll discuss matrix inverses and how to compute them in more detail later (Section 4.5). For now, it's important you know they exist.

The fundamental idea of linear algebra

In the remainder of the book, we'll learn all about the properties of vectors and matrices. Matrix-vector products play an important role in linear algebra because of their relation to *linear transformations*.

Functions are transformations from an input space (the domain) to an output space (the range). A linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ is a "vector function" that takes n-vectors as inputs and produces m-vectors as outputs. If the vector function T is linear, the output $\vec{y} = T(\vec{x})$ of T applied to $\vec{x}$ can be computed as the matrix-vector product $A_T\vec{x}$, for some matrix $A_T \in \mathbb{R}^{m\times n}$. We say T is *represented by* the matrix A_T. The matrix A_T is a particular "implementation" of the abstract linear transformation T. The coefficients of the matrix A_T depend on the basis for the input space and the basis for the output space.

Equivalently, every matrix $A \in \mathbb{R}^{m\times n}$ corresponds to some linear transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$. What does T_A do? We define the action of T_A on input $\vec{x}$ as the matrix-vector product $A\vec{x}$.

Given the equivalence between matrices and linear transformations, we can reinterpret the statement "linear algebra is about vectors and matrices" by saying "linear algebra is about vectors and linear transformations." If high school math is about numbers and functions, then linear algebra is about vectors and vector functions. The action of a function on a number is similar to the action of a linear transformation (matrix) on a vector:

$$
\begin{array}{rcl}
\text{function } f : \mathbb{R} \to \mathbb{R} & \Leftrightarrow & \text{linear transformation } T_A : \mathbb{R}^n \to \mathbb{R}^m \\
 & & \text{represented by the matrix } A \in \mathbb{R}^{m\times n} \\
\text{input } x \in \mathbb{R} & \Leftrightarrow & \text{input } \vec{x} \in \mathbb{R}^n \\
\text{output } f(x) \in \mathbb{R} & \Leftrightarrow & \text{output } T_A(\vec{x}) \equiv A\vec{x} \in \mathbb{R}^m \\
g \circ f\,(x) = g(f(x)) & \Leftrightarrow & T_B(T_A(\vec{x})) \equiv BA\vec{x} \\
\text{function inverse } f^{-1} & \Leftrightarrow & \text{matrix inverse } A^{-1} \\
\text{zeros of } f & \Leftrightarrow & \text{kernel of } T_A \equiv \text{null space of } A \equiv \mathcal{N}(A) \\
\text{image of } f & \Leftrightarrow & \text{image of } T_A \equiv \text{column space of } A \equiv \mathcal{C}(A)
\end{array}
$$

The above table of correspondences serves as a roadmap for the rest of the material in this book. There are several new concepts, but not too many. You can do this!

A good strategy is to try adapting your existing knowledge about functions to the world of linear transformations. For example, the zeros of a function $f(x)$ are the set of inputs for which the function's output is zero. Similarly, the *kernel* of a linear transformation T is the set of inputs that T sends to the zero vector. It's really the same concept; we're just upgrading functions to vector inputs.

In Chapter 1, I explained why functions are useful tools for modelling the real world. Well, linear algebra is the "vector upgrade" to your real-world modelling skills. With linear algebra you'll be able to model complex relationships between multivariable inputs and multivariable outputs. To build "modelling skills" you must first develop you geometrical intuition about lines, planes, vectors, bases, linear transformations, vector spaces, vector subspaces, etc. It's a lot of work, but the effort you invest will pay dividends.

Links

[Linear algebra lecture series by Prof. Strang from MIT]
`bit.ly/1ayRcrj` (original source of row and column pictures)

[A system of equations in the row picture and column picture]
`https://www.youtube.com/watch?v=uNKDw46_Ev4`

Exercises

What next?

Let's not get ahead of ourselves and bring geometry, vector spaces, algorithms, and the applications of linear algebra into the mix all at once. Instead, let's start with the basics. If linear algebra is about vectors and matrices, then we'd better define vectors and matrices precisely, and describe the math operations we can perform on them.

3.2 Review of vector operations

In Chapter 2 we described vectors from a practical point of view. Vectors are useful for describing directional quantities like forces and velocities in physics. In this section, we'll describe vectors more abstractly—as math objects. After defining a new mathematical object, the next step is to specify its properties and the operations we can perform on it.

Formulas

Consider the vectors $\vec{u} = (u_1, u_2, u_3)$ and $\vec{v} = (v_1, v_2, v_3)$, and an arbitrary constant $\alpha \in \mathbb{R}$. Vector algebra can be summarized as the following operations:

- $\alpha\vec{u} \equiv (\alpha u_1, \alpha u_2, \alpha u_3)$
- $\vec{u} + \vec{v} \equiv (u_1 + v_1, u_2 + v_2, u_3 + v_3)$

- $\vec{u} - \vec{v} \equiv (u_1 - v_1, u_2 - v_2, u_3 - v_3)$
- $||\vec{u}|| \equiv \sqrt{u_1^2 + u_2^2 + u_3^2}$
- $\vec{u} \cdot \vec{v} \equiv u_1 v_1 + u_2 v_2 + u_3 v_3$
- $\vec{u} \times \vec{v} \equiv (u_2 v_3 - u_3 v_2,\ u_3 v_1 - u_1 v_3,\ u_1 v_2 - u_2 v_1)$

In the next few pages we'll see what these operations can do for us.

Notation

The set of real numbers is denoted $\mathbb{R}$. An n-dimensional real vector consists of n real numbers slapped together in a bracket. We denote the set of 3-dimensional vectors as $(\mathbb{R}, \mathbb{R}, \mathbb{R}) \equiv \mathbb{R}^3$. Similarly, the set of n-dimensional real vectors is denoted $\mathbb{R}^n$.

Addition and subtraction

Addition and subtraction take pairs of vectors as inputs and produce vectors as outputs:

$$+ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \qquad \text{and} \qquad - : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n.$$

Addition and subtraction are performed component-wise:

$$\vec{w} = \vec{u} \pm \vec{v} \qquad \Leftrightarrow \qquad w_i = u_i \pm v_i, \quad \forall i \in [1, \ldots, n].$$

Scaling by a constant

Scaling is an operation that takes a number and a vector as inputs and produces a vector output:

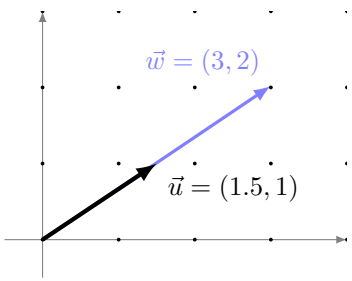
$$\text{scalar-mult} : \mathbb{R} \times \mathbb{R}^n \ \to \ \mathbb{R}^n.$$

There is no symbol to denote scalar multiplication—we just write the scaling factor in front of the vector and the multiplication is implicit.

The scaling factor α multiplying the vector $\vec{u}$ is equivalent to α multiplying each component of the vector:

$$\vec{w} = \alpha\vec{u} \qquad \Leftrightarrow \qquad w_i = \alpha u_i.$$

For example, choosing $\alpha = 2$, we obtain the vector $\vec{w} = 2\vec{u}$, which is two times longer than the vector $\vec{u}$:



$$\vec{w} = (w_1, w_2, w_3) = (2u_1, 2u_2, 2u_3) = 2(u_1, u_2, u_3) = 2\vec{u}.$$

Vector multiplication

The **dot product** takes pairs of vectors as inputs and produces real numbers as outputs:

$$\cdot : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \qquad \vec{u} \cdot \vec{v} \equiv \sum_{i=1}^{n} u_i v_i.$$

The dot product is defined for vectors of any dimension. As long as two vectors have the same length, we can compute their dot product.

The dot product is the key tool for projections, decompositions, and calculating orthogonality. It is also known as the *scalar product* or the *inner product*. Applying the dot product to two vectors produces a scalar number which carries information about *how similar* the two vectors are. Orthogonal vectors are not similar at all: no part of one vector goes in the same direction as the other vector, so their dot product is zero. For example, $\hat{\imath} \cdot \hat{\jmath} = 0$. Another notation for the inner product is $\langle \vec{u}, \vec{v} \rangle \equiv \vec{u} \cdot \vec{v}$.

The **cross product** takes pairs of three-dimensional vectors as inputs and produces three-dimensional vectors as outputs:

$$\times : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3, \qquad \vec{w} = \vec{u} \times \vec{v} \quad \Leftrightarrow \quad \begin{aligned} w_1 &= u_2 v_3 - u_3 v_2, \\ w_2 &= u_3 v_1 - u_1 v_3, \\ w_3 &= u_1 v_2 - u_2 v_1. \end{aligned}$$

The cross product, or *vector product* as it is sometimes called, is an operation that computes a vector that is perpendicular to both input vectors. For example: $\hat{\imath} \times \hat{\jmath} = \hat{k}$. Note the cross product is only defined for three-dimensional vectors.

Length of a vector

The *length* of the vector $\vec{u} \in \mathbb{R}^n$ is computed as follows:

$$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2} = \sqrt{\vec{u} \cdot \vec{u}}.$$

The length of a vector is a nonnegative number that describes the extent of the vector in space. The notion of length is an n-dimensional extension of Pythagoras' formula for the length of the hypotenuse in a right-angle triangle given the lengths of its two sides. The length of a vector is sometimes called the *magnitude* or the *norm* of the vector. The length of a vector $\vec{u}$ is denoted $\|\vec{u}\|$ or $|\vec{u}|_2$ or sometimes simply u.

There are many mathematical concepts that correspond to the intuitive notion of length. The formula above computes the *Euclid-*

ian length (or *Euclidian norm*) of the vector. Another name for the Euclidian length is the ℓ^2-norm (pronounced *ell-two norm*[2]).

Note a vector's length can be computed as the square root of the dot product of the vector with itself: $\|\vec{v}\| = \sqrt{\vec{v} \cdot \vec{v}}$. Indeed, there is a deep mathematical connection between norms and inner products.

Unit vectors

Given a vector $\vec{v}$ of any length, we can build a unit vector in the same direction by dividing $\vec{v}$ by its own length:

$$\hat{v} = \frac{\vec{v}}{||\vec{v}||}.$$

Unit vectors are useful in many contexts. When we want to specify a direction in space, we use a unit vector in that direction.

Projection

Okay, pop-quiz time! Let's see if you remember anything from Chapter 2. Suppose I give you a direction $\hat{d}$ and some vector $\vec{v}$, and ask you how much of $\vec{v}$ is in the direction $\hat{d}$? To find the answer, you must compute the dot product:

$$v_d = \hat{d} \cdot \vec{v} \equiv \|\hat{d}\|\|\vec{v}\| \cos\theta = 1\|\vec{v}\| \cos\theta,$$

where θ is the angle between $\vec{v}$ and $\hat{d}$. This formula is used in physics to compute x-components of forces: $F_x = \vec{F} \cdot \hat{\imath} = \|\vec{F}\| \cos\theta$.

Define the *projection* of a vector $\vec{v}$ in the $\hat{d}$ direction as follows:

$$\Pi_{\hat{d}}(\vec{v}) = v_d \hat{d} = (\hat{d} \cdot \vec{v})\hat{d}. \tag{3.1}$$

If the direction is specified by a vector $\vec{d}$ that is not of unit length, then the projection formula becomes

$$\Pi_{\vec{d}}(\vec{v}) = \left(\frac{\vec{d} \cdot \vec{v}}{\|\vec{d}\|^2}\right)\vec{d}. \tag{3.2}$$

Division by the length squared transforms the two appearances of the vector $\vec{d}$ into the unit vectors $\hat{d}$ needed for the projection formula:

$$\Pi_{\hat{d}}(\vec{v}) = \underbrace{(\vec{v} \cdot \hat{d})}_{\|\vec{v}\| \cos\theta} \hat{d} = \left(\vec{v} \cdot \frac{\vec{d}}{\|\vec{d}\|}\right)\frac{\vec{d}}{\|\vec{d}\|} = \left(\frac{\vec{v} \cdot \vec{d}}{\|\vec{d}\|^2}\right)\vec{d} = \Pi_{\vec{d}}(\vec{v}).$$

[2]The name ℓ^2-*norm* refers to the process of squaring each of the vector's components, and then taking the square root. Another norm is the ℓ^4-norm, defined as the fourth root of the sum of the vector's components raised to the fourth power: $|\vec{u}|_4 \equiv \sqrt[4]{u_1^4 + u_2^4 + u_3^4}$.

Remember these projection formulas well because we'll need to use them several times in this book: when computing projections onto planes (Section 5.2), when computing coordinates, and when describing the change-of-basis operation (Section 5.3).

Discussion

This section reviewed the properties of n-dimensional vectors, which are ordered tuples (lists) of n coefficients. It is important to think of vectors as whole mathematical objects and not as coefficients. Sure, all the vector operations boil down to manipulations of their coefficients, but vectors are most useful (and best understood) if you think of them as whole objects that have components, rather than focussing on their components.

Exercises

3.3 Matrix operations

A matrix is a two-dimensional array (a table) of numbers. Consider the m by n matrix $A \in \mathbb{R}^{m \times n}$. We denote the matrix as a whole A and refer to its individual entries as a_{ij}, where a_{ij} is the entry in the i^{th} row and the j^{th} column of A. What are the mathematical operations we can perform on this matrix?

Addition and subtraction

The matrix addition and subtraction operations take pairs of matrices as inputs and produce matrices as outputs:

$$+ : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n} \quad \text{and} \quad - : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}.$$

Addition and subtraction are performed component-wise:

$$C = A \pm B \quad \Leftrightarrow \quad c_{ij} = a_{ij} \pm b_{ij}, \ \forall i \in [1, \ldots, m], j \in [1, \ldots, n].$$

For example, addition for 3×2 matrices is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \\ a_{31} + b_{31} & a_{32} + b_{32} \end{bmatrix}.$$

Matrices must have the same dimensions to be added or subtracted.

Multiplication by a constant

Given a number α and a matrix A, we can *scale* A by α as follows:

$$\alpha A = \alpha \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} \\ \alpha a_{21} & \alpha a_{22} \\ \alpha a_{31} & \alpha a_{32} \end{bmatrix}.$$

Matrix-vector multiplication

The result of the matrix-vector product between a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $\vec{v} \in \mathbb{R}^n$ is an m-dimensional vector:

$$\text{matrix-vector product} : \mathbb{R}^{m \times n} \times \mathbb{R}^n \to \mathbb{R}^m$$

$$\vec{w} = A\vec{v} \qquad \Leftrightarrow \qquad w_i = \sum_{j=1}^{n} a_{ij} v_j, \quad \forall i \in [1, \ldots, m].$$

For example, the product of a 3×2 matrix A and the 2×1 column vector $\vec{v}$ results in a 3×1 vector:

$$\begin{aligned} A\vec{v} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= \underbrace{v_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + v_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}}_{\text{column picture}} \\ &= \left.\begin{bmatrix} (a_{11}, a_{12}) \cdot \vec{v} \\ (a_{21}, a_{22}) \cdot \vec{v} \\ (a_{31}, a_{32}) \cdot \vec{v} \end{bmatrix}\right\} \text{row picture} \\ &= \begin{bmatrix} a_{11}v_1 + a_{12}v_2 \\ a_{21}v_1 + a_{22}v_2 \\ a_{31}v_1 + a_{32}v_2 \end{bmatrix} \quad \in \mathbb{R}^{3 \times 1}. \end{aligned}$$

Note the two different ways to understand the matrix-vector product: the *column picture* and the *row picture*. In the column picture, the multiplication of the matrix A by the vector $\vec{v}$ is a **linear combination of the columns of the matrix**: $A\vec{v} = v_1 A_{[:,1]} + v_2 A_{[:,2]}$, where $A_{[:,1]}$ and $A_{[:,2]}$ are the two columns of the matrix A. In the row picture, multiplication of the matrix A by the vector $\vec{v}$ produces a column vector with coefficients equal to the **dot products of the rows of the matrix** A with the vector $\vec{v}$.

Matrix-matrix multiplication

The matrix product AB of matrices $A \in \mathbb{R}^{m \times \ell}$ and $B \in \mathbb{R}^{\ell \times n}$ consists of computing the dot product between each row of A and each column of B:

$$\text{matrix-product} : \mathbb{R}^{m \times \ell} \times \mathbb{R}^{\ell \times n} \to \mathbb{R}^{m \times n}$$

$$C = AB \qquad \Leftrightarrow \qquad c_{ij} = \sum_{k=1}^{\ell} a_{ik} b_{kj}, \forall i \in [1, \ldots, m], j \in [1, \ldots, n].$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} \end{bmatrix} \in \mathbb{R}^{3\times 2}.$$

Transpose

The transpose matrix A^T is defined by the formula $a_{ij}^\mathsf{T} = a_{ji}$. In other words, we obtain the transpose by "flipping" the matrix through its diagonal:

$$^\mathsf{T} : \mathbb{R}^{m\times n} \to \mathbb{R}^{n\times m},$$

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \alpha_3 & \beta_3 \end{bmatrix}.$$

Note that entries on the diagonal of the matrix are not affected by the transpose operation.

Properties of the transpose operation

- $(A + B)^\mathsf{T} = A^\mathsf{T} + B^\mathsf{T}$
- $(AB)^\mathsf{T} = B^\mathsf{T} A^\mathsf{T}$
- $(ABC)^\mathsf{T} = C^\mathsf{T} B^\mathsf{T} A^\mathsf{T}$
- $(A^\mathsf{T})^{-1} = (A^{-1})^\mathsf{T}$

Vectors as matrices

A vector is a special type of matrix. You can treat a vector $\vec{v} \in \mathbb{R}^n$ either as a *column vector* ($n \times 1$ matrix) or as a *row vector* ($1 \times n$ matrix).

Inner product

Recall the definition of the *dot product* or *inner product* for vectors:

$$\cdot : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \qquad \Leftrightarrow \qquad \vec{u} \cdot \vec{v} \equiv \sum_{i=1}^{n} u_i v_i.$$

If we think of these vectors as *column* vectors, we can write the dot product in terms of the matrix transpose operation $^\mathsf{T}$ and the standard

rules of matrix multiplication:

$$\vec{u} \cdot \vec{v} \equiv \vec{u}^{\mathsf{T}} \vec{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = u_1 v_1 + u_2 v_2 + u_3 v_3.$$

The dot product for vectors is really a special case of matrix multiplication. Alternatively, we could say that matrix multiplication is defined in terms of the dot product.

Outer product

Consider again two *column* vectors $\vec{u}$ and $\vec{v}$ ($n \times 1$ matrices). We obtain the inner product if we apply the transpose to the *first* vector in the product: $\vec{u}^{\mathsf{T}} \vec{v} \equiv \vec{u} \cdot \vec{v}$. If we instead apply the transpose to the *second* vector, we'll obtain the outer product of $\vec{u}$ and $\vec{v}$. The outer product operation takes pairs of vectors as inputs and produces matrices as outputs:

$$\text{outer-product} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}.$$

For example, the outer product of two vectors in $\mathbb{R}^3$ is

$$\vec{u}\vec{v}^{\mathsf{T}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{bmatrix} \qquad \in \mathbb{R}^{3 \times 3}.$$

Observe that the matrix-matrix product between a 3×1 matrix and a 1×3 matrix results in a 3×3 matrix.

In Section 5.2 we'll see how the outer product is used to build *projection matrices*. For example, the matrix that corresponds to the projection onto the x-axis is $M_x \equiv \hat{\imath}\hat{\imath}^{\mathsf{T}} \in \mathbb{R}^{3 \times 3}$. The x-projection of any vector $\vec{v}$ is computed as the matrix-vector product, $M_x \vec{v} = \hat{\imath}\hat{\imath}^{\mathsf{T}} \vec{v} = \hat{\imath}(\hat{\imath} \cdot \vec{v}) = v_x \hat{\imath}$.

Matrix inverse

Multiplying an invertible matrix A by its inverse A^{-1} produces the identity matrix: $AA^{-1} = \mathbb{1} = A^{-1}A$. The *identity matrix* obeys $\mathbb{1}\vec{v} = \vec{v}$ for all vectors $\vec{v}$. The inverse matrix A^{-1} *undoes* whatever A did. The cumulative effect of multiplying by A and A^{-1} is equivalent to the identity transformation,

$$A^{-1}(A(\vec{v})) = (A^{-1}A)\vec{v} = \mathbb{1}\vec{v} = \vec{v}.$$

We can think of "finding the inverse" $\text{inv}(A) = A^{-1}$ as an operation of the form,

$$\text{inv} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}.$$

Note that only *invertible* matrices have an inverse. Some matrices are not invertible—there is no "undo" operation for them. We'll postpone the detailed discussion of invertibility until Section 6.4.

Properties of matrix inverse operation

- $(A+B)^{-1} = A^{-1} + B^{-1}$
- $(AB)^{-1} = B^{-1}A^{-1}$
- $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$
- $(A^\mathsf{T})^{-1} = (A^{-1})^\mathsf{T}$

The matrix inverse plays the role of "division by the matrix A" in matrix equations. We'll discuss matrix equations in Section 4.2.

Trace

The *trace* of an $n \times n$ matrix is the sum of the n values on its diagonal:

$$\text{Tr} : \mathbb{R}^{n\times n} \to \mathbb{R}, \qquad \text{Tr}[A] \equiv \sum_{i=1}^{n} a_{ii}.$$

Properties of the trace operation

- $\text{Tr}[\alpha A + \beta B] = \alpha\text{Tr}[A] + \beta\text{Tr}[B]$ (linear property)
- $\text{Tr}[AB] = \text{Tr}[BA]$
- $\text{Tr}[ABC] = \text{Tr}[CAB] = \text{Tr}[BCA]$ (cyclic property)
- $\text{Tr}\big[A^\mathsf{T}\big] = \text{Tr}[A]$
- $\text{Tr}[A] = \sum_{i=1}^{n} \lambda_i$, where $\{\lambda_i\}$ are the eigenvalues of A

Determinant

The determinant of a matrix is a calculation that involves all the coefficients of the matrix, and whose output is a single number:

$$\det : \mathbb{R}^{n\times n} \to \mathbb{R}.$$

The determinant describes the relative geometry of the vectors that make up the rows of the matrix. More specifically, the determinant of a matrix A tells you the *volume* of a box with sides given by rows of A.

The determinant of a 2×2 matrix is

$$\det(A) = \det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

The quantity $ad - bc$ corresponds to the area of the parallelogram formed by the vectors (a, b) and (c, d). Observe that if the rows of A point in the same direction, $(a, b) = \alpha(c, d)$ for some $\alpha \in \mathbb{R}$, then the area of the parallelogram will be zero. If the determinant of a matrix is nonzero then the rows the matrix are linearly independent.

Properties of determinants

- $\det(AB) = \det(A)\det(B)$
- $\det(A) = \prod_{i=1}^{n} \lambda_i$, where $\{\lambda_i\}$ are the eigenvalues of A
- $\det\left(A^{\mathsf{T}}\right) = \det(A)$
- $\det\left(A^{-1}\right) = \frac{1}{\det(A)}$

Discussion

In the remainder of this book, you'll learn about various algebraic and geometric interpretations of the matrix operations we defined in this section. Understanding vector and matrix operations is essential for understanding more advanced theoretical topics and the applications of linear algebra. Seeing all these definitions at the same time can be overwhelming, I know, but the good news is that we've defined all the math operations and notation we'll use in the rest of the book. It could be worse, right?

So far, we've defined two of the main actors in linear algebra: vectors and matrices. But the introduction to linear algebra won't be complete until we introduce *linearity*. Linearity is the main thread that runs through all the topics in this book.

Exercises

3.4 Linearity

What is linearity and why do we need to spend an entire course learning about it? Consider the following arbitrary function that contains terms with different *powers* of the input variable x:

$$f(x) = \underbrace{a/x}_{\text{one-over-}x} + \underbrace{b}_{\text{constant}} + \underbrace{mx}_{\text{linear term}} + \underbrace{qx^2}_{\text{quadratic}} + \underbrace{cx^3}_{\text{cubic}}.$$

The term mx is the *linear* term in this expression—it contains x to the first power. All other terms are *nonlinear*. In this section we'll discuss the properties of expressions containing only linear terms.

Introduction

A single-variable function takes as input a real number x and outputs a real number y. The signature of this class of functions is

$$f : \mathbb{R} \to \mathbb{R}.$$

The most general *linear* function from $\mathbb{R}$ to $\mathbb{R}$ looks like this:

$$y \equiv f(x) = mx,$$

where $m \in \mathbb{R}$ is called the *coefficient* of x. The action of a linear function is to multiply the input by the constant m. So far so good.

Example of composition of linear functions Given the linear functions $f(x) = 2x$ and $g(y) = 3y$, what is the equation of the function $h(x) \equiv g \circ f(x) = g(f(x))$? The composition of the functions $f(x) = 2x$ and $g(y) = 3y$ is the function $h(x) = g(f(x)) = 3(2x) = 6x$. Note the composition of two linear functions is also a linear function. The coefficient of h is equal to the product of the coefficients of f and g.

Definition

A function f is *linear* if it satisfies the equation

$$f(\alpha x_1 + \beta x_2) = \alpha f(x_1) + \beta f(x_2),$$

for any two inputs x_1 and x_2 and all constants α and β. Linear functions map a linear combination of inputs to the same linear combination of outputs.

Lines are not linear functions!

Consider the equation of a line:

$$l(x) = mx + b,$$

where the constant m corresponds to the slope of the line, and the constant $b \equiv f(0)$ is the y-intercept of the line. A *line* $l(x) = mx + b$ with $b \neq 0$ is *not* a linear function. This logic may seem a bit weird, but if you don't trust me, you can check for yourself:

$$l(\alpha x_1 + \beta x_2) = m(\alpha x_1 + \beta x_2) + b \neq m(\alpha x_1) + b + m(\beta x_2) + b = \alpha l(x_1) + \beta l(x_2).$$

A function with a linear part plus a constant term is called an *affine transformation*. Affine transformations are cool but a bit off-topic, since the focus in this book will be on *linear* transformations.

Multivariable functions

The study of linear algebra is the study of *all* things linear. In particular, we'll learn how to work with functions that take multiple variables as inputs. Consider the set of functions that take pairs of real numbers as inputs and produce real numbers as outputs:

$$f\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}.$$

The most general linear function of two variables is

$$f(x, y) = m_x x + m_y y.$$

You can think of m_x as the x-slope and m_y as the y-slope of the function. We say m_x is the x-coefficient and m_y the y-coefficient in the linear expression $m_x x + m_y y$.

Linear expressions

A *linear expression* in the variables x_1, x_2, and x_3 has the form,

$$a_1 x_1 + a_2 x_2 + a_3 x_3,$$

where a_1, a_2, and a_3 are arbitrary constants. Note the new terminology, "`expr` is linear in v," which refers to an expression in which the variable v is raised to the first power. The expression $\frac{1}{a} x_1 + b^6 x_2 + \sqrt{c}\, x_3$ contains nonlinear factors ($\frac{1}{a}$, b^6, and $\sqrt{c}$) but is a linear expression in the variables x_1, x_2, and x_3.

Linear equations

A linear equation in the variables x_1, x_2, and x_3 has the form

$$a_1 x_1 + a_2 x_2 + a_3 x_3 = c.$$

This equation is linear because it contains only linear terms in the variables x_1, x_2, and x_3.

Example Linear equations are very versatile. Suppose you know the following equation describes some real-world phenomenon:

$$4k - 2m + 8p = 10,$$

where k, m, and p correspond to three variables of interest. You can interpret this equation as describing the variable m as a function of the variables k and p, and rewrite the equation as

$$m(k, p) = 2k + 4p - 5.$$

Using this function, you can predict the value of m given the knowledge of the quantities k and p.

Another option would be to interpret k as a function of m and p: $k(m,p) = \frac{10}{4} + \frac{m}{2} - 2p$. This model would be useful if you know the quantities m and p and want to predict the value of the variable k.

Applications

Geometrical interpretation of linear equations

The linear equation in x and y,

$$ax + by = c, \qquad b \neq 0,$$

corresponds to a line with the equation $y(x) = mx + y_0$ in the Cartesian plane. The slope of the line is $m = -a/b$ and its y-intercept is $y_0 = c/b$. The special case when $b = 0$ corresponds to a vertical line with equation $x = \frac{c}{a}$.

The most general linear equation in x, y, and z,

$$ax + by + cz = d,$$

corresponds to the equation of a plane in a three-dimensional space. Assuming $c \neq 0$, we can rewrite this equation so z (the "height") is a function of the coordinates x and y: $z(x,y) = z_0 + m_x x + m_y y$. The slope of the plane in the x-direction is $m_x = -\frac{a}{c}$ while the slope in the y-direction is $m_y = -\frac{b}{c}$. The z-intercept of this plane is $z_0 = \frac{d}{c}$.

First-order approximations

When we use a linear function as a mathematical model for a nonlinear, real-world input-output process, we say the function represents a *linear model* or a *first-order approximation* for the process. Let's analyze what this means in more detail, and see why linear models are so popular in science.

In calculus, we learn that functions can be represented as infinite Taylor series:

$$f(x) = \text{taylor}(f(x)) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots = \sum_{n=0}^{\infty} a_n x^n,$$

where the coefficient a_n depends on the n^{th} derivative of $f(x)$. The Taylor series is only equal to the function $f(x)$ if infinitely many terms in the series are calculated. If we sum together only a finite number of terms in the series, we obtain a *Taylor series approximation*. The first-order Taylor series approximation to $f(x)$ is

$$f(x) \approx \text{taylor}_1(f(x)) = a_0 + a_1 x = f(0) + f'(0)x.$$

The above equation describes the best approximation to $f(x)$ near $x = 0$, by a line of the form $l(x) = mx+b$. To build a linear model $f(x)$ of a real-world process, it is sufficient to measure two parameters: the initial value $b \equiv f(0)$ and the rate of change $m \equiv f'(0)$.

Scientists routinely use linear models because this kind of model allows for easy *parametrization*. To build a linear model, the first step is to establish the initial value $f(0)$ by inputting $x = 0$ to the process and seeing what comes out. Next, we vary the input by some amount Δx and observe the resulting change in the output Δf. The rate of change parameter is equal to the change in the output divided by the change in the input $m = \frac{\Delta f}{\Delta x}$. Thus, we can obtain the parameters of a linear model in two simple steps. In contrast, finding the parametrization of nonlinear models is a more complicated task.

For a function $F(x, y, z)$ that takes three variables as inputs, the first-order Taylor series approximation is

$$F(x, y, z) \approx b + m_x x + m_y y + m_z z.$$

Except for the constant term, the function has the form of a linear expression. The "first-order approximation" to a function of n variables $F(x_1, x_2, \ldots, x_n)$ has the form $b + m_1 x_1 + m_2 x_2 + \cdots + m_n x_n$.

As in the single-variable case, finding the parametrization of a multivariable linear model is a straightforward task. Suppose we want to model some complicated real-world phenomenon that has n input variables. First, we input only zeros to obtain the initial value $F(0, \ldots, 0) \equiv b$. Next, we go through each of the input variables one-by-one and measure how a small change in each input Δx_i affects the output Δf. The rate of change with respect to the input x_i is $m_i = \frac{\Delta f}{\Delta x_i}$. By combining the knowledge of the initial value b and the "slopes" with respect to each input parameter, we'll obtain a complete linear model of the phenomenon.

Discussion

In the next three chapters, we'll learn about new mathematical objects and mathematical operations. Linear algebra is the study vectors, matrices, linear transformations, vector spaces, and more abstract vector-like objects. The mathematical operations we'll perform on these objects will be linear: $f(\alpha\mathbf{obj}_1+\beta\mathbf{obj}_2) = \alpha f(\mathbf{obj}_1)+\beta f(\mathbf{obj}_2)$. Linearity is the core assumption of linear algebra.

Exercises

E3.1 Are these expressions linear in the variables x, y, and z?

a) $2x+5y+\sqrt{m}z$ **b)** $10\sqrt{x}+2(y+z)$ **c)** $42x+\alpha^2\sin(\frac{\pi}{3})y+z\cos(\frac{\pi}{3})$

3.5 Overview of linear algebra

In linear algebra, you'll learn new computational techniques and develop new ways of thinking about math. With these new tools, you'll be able to use linear algebra techniques for many applications. Let's now look at what lies ahead in the book.

Computational linear algebra

The first steps toward understanding linear algebra will seem a little tedious. In Chapter 4 you'll develop basic skills for manipulating vectors and matrices. Matrices and vectors have many components and performing operations on them involves many arithmetic steps—there is no way to circumvent this complexity. Make sure you understand the basic algebra rules (how to add, subtract, and multiply vectors and matrices) because they are a prerequisite for learning more advanced material.

The good news is, with the exception of your homework assignments and final exam, you won't have to carry out matrix algebra by hand. It is much more convenient to use a computer for large matrix calculations. For small instances, like 4×4 matrices, you should be able to perform all the matrix algebra operations with pen and paper. The more you develop your matrix algebra skills, the deeper you'll be able to delve into the advanced topics.

Geometrical linear algebra

So far, we described vectors and matrices as arrays of numbers. This is fine for the purpose of doing *algebra* on vectors and matrices, but this description is not sufficient for understanding their geometrical properties. The components of a vector $\vec{v} \in \mathbb{R}^n$ can be thought of as measuring distances along a coordinate system with n axes. The vector $\vec{v}$ can therefore be said to "point" in a particular direction with respect to the coordinate system. The fun part of linear algebra starts when you learn about the geometrical interpretation of the algebraic operations on vectors and matrices.

Consider some unit length vector that specifies a direction of interest $\hat{r}$. Suppose we're given some other vector $\vec{v}$, and we're asked to find *how much of* $\vec{v}$ *is in the* $\hat{r}$ *direction.* The answer is computed using the dot product: $v_r = \vec{v} \cdot \hat{r} = \|\vec{v}\| \cos\theta$, where θ is the angle between $\vec{v}$ and $\hat{r}$. The technical term for the quantity v_r is "the length of the projection of $\vec{v}$ in the $\hat{r}$ direction." By "projection," I mean we

ignore all parts of $\vec{v}$ that are not in the $\hat{r}$ direction. Projections are used in mechanics to calculate the x- and y-components of forces in force diagrams. In Section 5.2 we'll learn how to calculate all kinds of projections using the dot product.

To further consider the geometrical aspects of vector operations, imagine the following situation. Suppose I gave you two vectors $\vec{u}$ and $\vec{v}$, and asked you to find a third vector $\vec{w}$ that is perpendicular to both $\vec{u}$ and $\vec{v}$. A priori this sounds like a complicated question to answer, but in fact the required vector $\vec{w}$ can easily be obtained by computing the cross product $\vec{w} = \vec{u} \times \vec{v}$.

In Section 5.1 we'll learn how to describe lines and planes in terms of points, direction vectors, and normal vectors. Consider the following geometric problem: given the equations of two planes in $\mathbb{R}^3$, find the equation of the line where the two planes intersect. There is an algebraic procedure called *Gauss–Jordan elimination* you can use to find the solution.

The determinant of a matrix has a geometrical interpretation (Section 4.4). The determinant tells us something about the relative orientation of the vectors that make up the rows of the matrix. If the determinant of a matrix is zero, it means the rows are not *linearly independent*, in other words, at least one of the rows can be written in terms of the other rows. Linear independence, as we'll see shortly, is an important property for vectors to have. The determinant is a convenient way to test whether a set of vectors are linearly independent.

It is important that you *visualize* each new concept you learn about. Always keep a picture in your head of what is going on. The relationships between two-dimensional vectors can be represented in vector diagrams. Three-dimensional vectors can be visualized by pointing pens and pencils in different directions. Most of the intuitions you build about vectors in two and three dimensions are applicable to vectors with more dimensions.

Theoretical linear algebra

Linear algebra will teach you how to reason about vectors and matrices in an abstract way. By thinking abstractly, you'll be able to extend your geometrical intuition of two and three-dimensional problems to problems in higher dimensions. Much *knowledge buzz* awaits you as you learn about new mathematical ideas and develop new ways of thinking.

You're no doubt familiar with the normal coordinate system made of two orthogonal axes: the x-axis and the y-axis. A vector $\vec{v} \in \mathbb{R}^2$ is specified in terms of its coordinates (v_x, v_y) with respect to these

axes. When we say $\vec{v} = (v_x, v_y)$, what we really mean is $\vec{v} = v_x\hat{\imath} + v_y\hat{\jmath}$, where $\hat{\imath}$ and $\hat{\jmath}$ are unit vectors that point along the x- and y-axes. As it turns out, we can use many other kinds of coordinate systems to represent vectors. A *basis* for $\mathbb{R}^2$ is any set of two vectors $\{\hat{e}_1, \hat{e}_2\}$ that allows us to express all vectors $\vec{v} \in \mathbb{R}^2$ as linear combinations of the basis vectors: $\vec{v} = v_1\hat{e}_1 + v_2\hat{e}_2$. The same vector $\vec{v}$ corresponds to two different coordinate pairs, depending on which basis is used for the description: $\vec{v} = (v_x, v_y)$ in the basis $\{\hat{\imath}, \hat{\jmath}\}$ and $\vec{v} = (v_1, v_2)$ in the basis $\{\hat{e}_1, \hat{e}_2\}$. We'll learn about bases and their properties in great detail in the coming chapters. The choice of basis plays a fundamental role in all aspects of linear algebra. Bases relate the real-world to its mathematical representation in terms of vector and matrix components.

In the text above, I explained that computing the product between a matrix and a vector $A\vec{x} = \vec{y}$ can be thought of as a linear vector function, with input $\vec{x}$ and output $\vec{y}$. Any linear transformation (Section 6.1) can be represented (Section 6.2) as a multiplication by a matrix A. Conversely, every $m \times n$ matrix $A \in \mathbb{R}^{m\times n}$ can be thought of as implementing some linear transformation (vector function): $T_A \colon \mathbb{R}^n \to \mathbb{R}^m$. The equivalence between matrices and linear transformations allows us to identify certain matrix properties with properties of linear transformations. For example, the *column space* $\mathcal{C}(A)$ of the matrix A (the set of vectors that can be written as a combination of the columns of A) corresponds to the image space of the linear transformation T_A (the set of possible outputs of T_A).

The eigenvalues and eigenvectors of matrices (Section 7.1) allow us to describe the actions of matrices in a natural way. The set of eigenvectors of a matrix are special input vectors for which the action of the matrix is described as a *scaling*. When a matrix acts on one of its eigenvectors, the output is a vector in the same direction as the input vector scaled by a constant. The scaling constant is the *eigenvalue* (own value) associated this eigenvector. By specifying all the eigenvectors and eigenvalues of a matrix, it is possible to obtain a complete description of what the matrix does. Thinking of matrices in term of their eigenvalues and eigenvectors is a very powerful technique for describing their properties and has many applications.

Part of what makes linear algebra so powerful is that linear algebra techniques can be applied to all kinds of "vector-like" objects. The abstract concept of a vector space (Section 7.3) captures precisely what it means for some class of mathematical objects to be "vector-like." For example, the set of polynomials of degree at most two, denoted $P_2(x)$, consists of all functions of the form $f(x) = a_0 + a_1x + a_2x^2$. Polynomials are vector-like because it's possible to describe each polynomial in terms of its coefficients (a_0, a_1, a_2). Furthermore,

the sum of two polynomials and the multiplication of a polynomial by a constant both correspond to vector-like calculations of coefficients. Once you realize polynomials are vector-like, you'll be able to use linear algebra concepts like *linear independence*, *dimension*, and *basis* when working with polynomials.

Useful linear algebra

One of the most useful skills you'll learn in linear algebra is the ability to solve systems of linear equations. Many real-world problems are expressed as linear equations in multiple unknown quantities. You can solve for n unknowns simultaneously if you have a set of n linear equations that relate the unknowns. To solve this system of equations, you can use basic techniques such as substitution, subtraction, and elimination by equating to eliminate the variables one by one (see Section 1.15), but the procedure will be slow and tedious for many unknowns. If the system of equations is linear, it can be expressed as an *augmented matrix* built from the coefficients in the equations. You can then use the Gauss–Jordan elimination algorithm to solve for the n unknowns (Section 4.1). The key benefit of the augmented matrix approach is that it allows you to focus on the coefficients without worrying about the variable names. This saves time when you must solve for many unknowns. Another approach for solving n linear equations in n unknowns is to express the system of equations as a matrix equation (Section 4.2) and then solve the matrix equation by computing the matrix inverse (Section 4.5).

In Section 7.6 you'll learn how to *decompose* a matrix into a product of simpler matrices. Matrix decompositions are often performed for computational reasons: certain problems are easier to solve on a computer when the matrix is expressed in terms of its simpler constituents. Other decompositions, like the decomposition of a matrix into its eigenvalues and eigenvectors, give you valuable information about the properties of the matrix. Google's original `PageRank` algorithm for ranking webpages by "importance" can be explained as the search for an eigenvector of a matrix. The matrix in question contains information about all hyperlinks that exist between webpages. The eigenvector we're looking for corresponds to a vector that describes the relative importance of each page. So when I tell you eigenvectors are *valuable information*, I am not kidding: a 350-billion dollar company started as an eigenvector idea.

The techniques of linear algebra find applications in many areas of science and technology. We'll discuss applications such as *modelling* multidimensional real-world problems, finding *approximate solutions* to equations (curve fitting), solving constrained optimization prob-

lems using *linear programming*, and many other in Chapter 8. As a special bonus for readers interested in physics, a short introduction to quantum mechanics can be found in Chapter 10; if you have a good grasp of linear algebra, you can understand matrix quantum mechanics at no additional mental cost.

Our journey of all things linear begins with the computational aspects of linear algebra. In Chapter 4 we'll learn how to efficiently solve large systems of linear equations, practice computing matrix products, discuss matrix determinants, and compute matrix inverses.

3.6 Introductory problems

Before we continue with the new material in linear algebra, we want to make sure you got the definitions down straight.

linearity, vector operations, matrix operations, etc.

P3.1 Find the sum of the vectors $(1, 0, 1)$ and the vector $(0, 2, 2)$.

Hint: Vector addition is performed element-wise.

P3.2 You friend is taking a physics class and needs some help with a vector question. Can you help your friend answer this question: Let $|a\rangle = 1|0\rangle + 3|1\rangle$ and $|b\rangle = 4|0\rangle - 1|1\rangle$. Find $|a\rangle + |b\rangle$.

Hint: The weird angle-bracket notation denotes basis vectors: $|x\rangle \equiv \vec{e}_x$.

P3.3 Given $\vec{v} = (2, -1, 3)$ and $\vec{w} = (1, 0, 1)$, compute the following vector products: a) $\vec{v} \cdot \vec{w}$; b) $\vec{v} \times \vec{w}$; c) $\vec{v} \times \vec{v}$; d) $\vec{w} \times \vec{w}$.

P3.4 Given unit vectors $\hat{\imath} = (1, 0, 0)$, $\hat{\jmath} = (0, 1, 0)$ and $\hat{k} = (0, 0, 1)$. Find the following cross products: a) $\hat{\imath} \times \hat{\imath}$; b) $\hat{\imath} \times \hat{\jmath}$; c) $(-\hat{\imath}) \times \hat{k} + \hat{\jmath} \times \hat{\imath}$; d) $\hat{k} \times \hat{\jmath} + \hat{\imath} \times \hat{\imath} + \hat{\jmath} \times \hat{k} + \hat{\jmath} \times \hat{\imath}$.

Chapter 4

Computational linear algebra

This chapter covers the computational aspects of performing matrix calculations. Understanding matrix computations is important because all later chapters depend on them. Suppose we're given a huge matrix $A \in \mathbb{R}^{n \times n}$ with $n = 1000$. Hidden behind the innocent-looking mathematical notation of the matrix inverse A^{-1}, the matrix product AA, and the matrix determinant $|A|$, lie monster computations involving all the $1000 \times 1000 = 1$ million entries of the matrix A. Millions of arithmetic operations must be performed... so I hope you have at least a thousand pencils ready!

Okay, calm down. I won't *actually* make you calculate millions of arithmetic operations. In fact, to learn linear algebra, it is sufficient to know how to carry out calculations with 3×3 and 4×4 matrices. Even for such moderately sized matrices, computing products, inverses, and determinants by hand are serious computational tasks. If you're ever required to take a linear algebra final exam, you need to make sure you can do these calculations quickly. Even if no exam looms in your imminent future, it's important to practice matrix operations by hand to get a feel for them.

This chapter will introduce you to the following computational tasks involving matrices:

Gauss–Jordan elimination Suppose we're trying to solve two equations in two unknowns x and y:

$$\begin{aligned} ax + by &= c, \\ dx + ey &= f. \end{aligned}$$

If we add α-times the first equation to the second equation, we obtain an equivalent system of equations:

$$\begin{aligned} ax \;+\; & by \;=\; c \\ (d+\alpha a)x + (e+\alpha b)y &= f + \alpha c. \end{aligned}$$

This is called a *row operation*: we added α-times the first row to the second row. Row operations change the coefficients of the system of equations, but leave the solution unchanged. Gauss–Jordan elimination is a systematic procedure for solving systems of linear equations using row operations.

Matrix product The product AB between matrices $A \in \mathbb{R}^{m\times\ell}$ and $B \in \mathbb{R}^{\ell\times n}$ is the matrix $C \in \mathbb{R}^{m\times n}$ whose coefficients c_{ij} are defined by the formula $c_{ij} = \sum_{k=1}^{\ell} a_{ik}b_{kj}$ for all $i \in [1,\ldots,m]$ and $j \in [1,\ldots,n]$. In Section 4.3 we'll unpack this formula and learn about its intuitive interpretation: that computing $C = AB$ is computing all the dot products between the rows of A and the columns of B.

Determinant The determinant of a matrix A, denoted $|A|$, is an operation that gives us useful information about the linear independence of the rows of the matrix. The determinant is connected to many notions of linear algebra: linear independence, geometry of vectors, solving systems of equations, and matrix invertibility. We'll discuss these aspects of determinants in Section 4.4.

Matrix inverse In Section 4.5 we'll build upon our knowledge of Gauss–Jordan elimination, matrix products, and determinants to derive three different procedures for computing the matrix inverse A^{-1}.

4.1 Reduced row echelon form

In this section we'll learn to solve systems of linear equations using the *Gauss–Jordan elimination* procedure. A system of equations can be represented as a matrix of coefficients. The Gauss–Jordan elimination procedure converts any matrix into its *reduced row echelon form* (RREF). We can easily find the solution (or solutions) of the system of equations from the RREF.

Listen up: the material covered in this section requires your full-on, caffeinated attention, as the procedures you'll learn are somewhat tedious. Gauss–Jordan elimination involves many repetitive mathematical manipulations of arrays of numbers. It's important you hang in there and follow through the step-by-step manipulations, as well

as verify each step I present *on your own* with pen and paper. Don't just take my word for it—always verify the steps!

Solving equations

Suppose you're asked to solve the following system of equations:

$$\begin{aligned} 1x_1 + 2x_2 &= 5 \\ 3x_1 + 9x_2 &= 21. \end{aligned}$$

The standard approach is to use one of the equation-solving tricks we learned in Section 1.15 to combine the equations and find the values of the two unknowns x_1 and x_2.

Observe that the *names* of the two unknowns are irrelevant to the solution of the system of equations. Indeed, the solution (x_1, x_2) to the above system of equations is the same as the solution (s, t) to the system of equations

$$\begin{aligned} 1s + 2t &= 5 \\ 3s + 9t &= 21. \end{aligned}$$

The important parts of a system of linear equations are the *coefficients* in front of the variables and the constants on the right-hand side of each equation.

Augmented matrix

The system of linear equations can be written as an *augmented matrix*:

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 3 & 9 & 21 \end{array}\right].$$

The first column corresponds to the coefficients of the first variable, the second column is for the second variable, and the last column corresponds to the constants of the right-hand side. It is customary to draw a vertical line where the equal signs in the equations would normally appear. This line helps distinguish the coefficients of the equations from the column of constants on the right-hand side.

Once we have the augmented matrix, we can simplify it by using *row operations* (which we'll discuss shortly) on its entries. After simplification by row operations, the augmented matrix will be transformed to

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right],$$

which corresponds to the system of equations

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2. \end{aligned}$$

This is a *trivial* system of equations; there is nothing left to solve and we can see the solutions are $x_1 = 1$ and $x_2 = 2$. This example illustrates the general idea of the Gauss–Jordan elimination procedure for solving systems of equations by manipulating an augmented matrix.

Row operations

We can manipulate the rows of an augmented matrix without changing its solutions. We're allowed to perform the following three types of row operations:

- Add a multiple of one row to another row
- Swap the position of two rows
- Multiply a row by a constant

Let's trace the sequence of row operations needed to solve the system of equations

$$\begin{aligned} 1x_1 + 2x_2 &= 5 \\ 3x_1 + 9x_2 &= 21, \end{aligned}$$

starting from its augmented matrix:

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 3 & 9 & 21 \end{array}\right].$$

1. As a first step, we eliminate the first variable in the second row by subtracting three times the first row from the second row:

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 0 & 3 & 6 \end{array}\right].$$

   We denote this row operation as $R_2 \leftarrow R_2 - 3R_1$.

2. To simplify the second row, we divide it by 3 to obtain

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 0 & 1 & 2 \end{array}\right].$$

   This row operation is denoted $R_2 \leftarrow \frac{1}{3}R_2$.

3. The final step is to eliminate the second variable from the first row. We do this by subtracting two times the second row from the first row, $R_1 \leftarrow R_1 - 2R_2$:

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right].$$

We can now read off the solution: $x_1 = 1$ and $x_2 = 2$.

Note how we simplified the augmented matrix through a specific procedure: we followed the *Gauss–Jordan elimination* algorithm to bring the matrix into its *reduced row echelon form*.

The *reduced row echelon form* (RREF) is the simplest form for an augmented matrix. Each row contains a *leading one* (a numeral 1) also known as a *pivot*. Each column's pivot is used to eliminate the numbers that lie below and above it in the same column. The end result of this procedure is the reduced row echelon form:

$$\left[\begin{array}{cccc|c} 1 & 0 & * & 0 & * \\ 0 & 1 & * & 0 & * \\ 0 & 0 & 0 & 1 & * \end{array}\right].$$

Note the matrix contains only zero entries below and above the pivots. The asterisks $*$ denote arbitrary numbers that could not be eliminated because no leading one is present in these columns.

Definitions

- The *solution* to a system of linear equations in the variables $x_1, x_2, \ldots, x_n$ is the set of values $\{(x_1, x_2, \ldots, x_n)\}$ that satisfy *all* the equations.
- The *pivot* for row j of a matrix is the left-most nonzero entry in the row j. Any *pivot* can be converted into a *leading one* by an appropriate scaling of that row.
- *Gaussian elimination* is the process of bringing a matrix into *row echelon form*.
- A matrix is said to be in *row echelon form* (REF) if all entries below the leading ones are zero. This form can be obtained by adding or subtracting the row with the leading one from the rows below it.
- *Gaussian-Jordan elimination* is the process of bringing a matrix into *reduced row echelon form*.
- A matrix is said to be in *reduced row echelon form* (RREF) if all the entries below *and above* the pivots are zero. Starting from the REF, we obtain the RREF by subtracting the row containing the pivots from the rows above them.

- rank(A): the *rank* of the matrix A is the number of pivots in the RREF of A.

Gauss–Jordan elimination algorithm

The Gauss–Jordan elimination algorithm proceeds in two phases: a forward phase in which we move left to right, and a backward phase in which we move right to left.

1. Forward phase (left to right):
   - 1.1 Obtain a pivot (a leading one) in the leftmost column.
   - 1.2 Subtract the row with the pivot from all rows below it to obtain zeros in the entire column.
   - 1.3 Look for a leading one in the next column and repeat.
2. Backward phase (right to left):
   - 2.1 Find the rightmost pivot and use it to eliminate all numbers above the pivot in its column.
   - 2.2 Move one column to the left and repeat.

Example We're asked to solve the following system of equations:

$$\begin{aligned} 1x + 2y + 3z &= 14 \\ 2x + 5y + 6z &= 30 \\ -1x + 2y + 3z &= 12. \end{aligned}$$

The first step toward the solution is to build the augmented matrix that corresponds to this system of equations:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 2 & 5 & 6 & 30 \\ -1 & 2 & 3 & 12 \end{array}\right].$$

We can now start the left-to-right phase of the algorithm:

1. Conveniently, there is a leading one at the top of the leftmost column. If a zero were there instead, a row-swap operation would be necessary to obtain a nonzero entry.
2. The next step is to clear the entries in the entire column below this pivot. The row operations we'll use for this purpose are $R_2 \leftarrow R_2 - 2R_1$ and $R_3 \leftarrow R_3 + R_1$:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 0 & 1 & 0 & 2 \\ 0 & 4 & 6 & 26 \end{array}\right].$$

3. We now shift our attention to the second column. Using the leading one for the second column, we set the number in the column below it to zero using $R_3 \leftarrow R_3 - 4R_2$. The result is

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 6 & 18 \end{array}\right].$$

4. Next, we move to the third column. Instead of a leading one, we find it contains a "leading six," which we can convert to a leading one using $R_3 \leftarrow \frac{1}{6}R_3$. We thus obtain

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right].$$

The forward phase of the Gauss–Jordan elimination procedure is now complete. We identified three pivots and used them to systematically set all entries below each pivot to zero. The matrix is now in *row echelon form*.

Next we start the backward phase of the Gauss–Jordan elimination procedure, during which we'll work right-to-left to set all numbers above each pivot to zero:

5. The first row operation is $R_1 \leftarrow R_1 - 3R_3$, and it leads to

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right].$$

6. The final step is $R_1 \leftarrow R_1 - 2R_2$, which gives

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right].$$

The matrix is now in *reduced row echelon form* and we can see the solution is $x = 1$, $y = 2$, and $z = 3$.

We've described the general idea of the Gauss–Jordan elimination and explored some examples where the solutions to the system of equations were *unique*. There are other possibilities for the solutions of a system of linear equations. We'll describe these other possible scenarios next.

Number of solutions

A system of three linear equations in three variables could have:

- **One solution**. If the RREF of a matrix has a pivot in each row, we can read off the values of the solution by inspection:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & c_1 \\ 0 & 1 & 0 & c_2 \\ 0 & 0 & 1 & c_3 \end{array}\right].$$

  The *unique* solution is $x_1 = c_1$, $x_2 = c_2$, and $x_3 = c_3$.

- **Infinitely many solutions 1**. If one of the equations is redundant, a row of zeros will appear when the matrix is brought to the RREF. This happens when one of the original equations is a linear combination of the other two. In such cases, we're really solving *two* equations in *three* variables, so can't "pin down" one of the unknown variables. We say the solution contains a *free variable*. For example, consider the following RREF:

$$\left[\begin{array}{ccc|c} 1 & 0 & a_1 & c_1 \\ 0 & 1 & a_2 & c_2 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

  The column that doesn't contain a leading one corresponds to the free variable. To indicate that x_3 is a free variable, we give it a special label, $x_3 \equiv t$. The variable t could be any number $t \in \mathbb{R}$. In other words, when we say t is free, it means t can take on *any* value from $-\infty$ to $+\infty$. The information in the augmented matrix can now be used to express x_1 and x_2 in terms of the right-hand constants and the free variable t:

$$\left\{\begin{array}{l} x_1 = c_1 - a_1\,t \\ x_2 = c_2 - a_2\,t \\ x_3 = t \end{array}\ ,\quad \forall t \in \mathbb{R}\right\} = \left\{\begin{bmatrix} c_1 \\ c_2 \\ 0 \end{bmatrix} + t\begin{bmatrix} -a_1 \\ -a_2 \\ 1 \end{bmatrix},\quad \forall t \in \mathbb{R}\right\}.$$

  The solution corresponds to the equation of a line passing through the point $(c_1, c_2, 0)$ with direction vector $(-a_1, -a_2, 1)$. We'll discuss the geometry of lines in Chapter 5. For now, it's important you understand that a system of equations can have more than one solution; any point on the line $\ell \equiv \{(c_1, c_2, 0) + t(-a_1, -a_2, 1),\ \forall t \in \mathbb{R}\}$ is a solution to the above system of equations.

- **Infinitely many solutions 2**. It's also possible to obtain a two-dimensional solution space. This happens when two of the

three equations are redundant. In this case, there will be a single leading one, and thus two free variables. For example, in the RREF

$$\left[\begin{array}{ccc|c} 0 & 1 & a_2 & c_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right],$$

the variables x_1 and x_3 are free. As in the previous infinitely-many-solutions case, we define new labels for the free variables $x_1 \equiv s$ and $x_3 \equiv t$, where $s \in \mathbb{R}$ and $t \in \mathbb{R}$ are two arbitrary numbers. The solution to this system of equations is

$$\left\{\begin{array}{l} x_1 = s \\ x_2 = c_2 - a_2 t, \quad \forall s,t \in \mathbb{R} \\ x_3 = t \end{array}\right\} = \left\{\begin{bmatrix} 0 \\ c_2 \\ 0 \end{bmatrix} + s\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + t\begin{bmatrix} 0 \\ -a_2 \\ 1 \end{bmatrix},\ \forall s,t \in \mathbb{R}\right\}.$$

This solution set corresponds to the *parametric equation* of a plane that contains the point $(0, c_2, 0)$ and the vectors $(1, 0, 0)$ and $(0, -a_2, 1)$.

The *general equation* for the solution plane is $0x+1y+a_2z = c_2$, as can be observed from the first row of the augmented matrix. In Section 5.1 we'll learn more about the geometry of planes and how to convert between their general and parametric forms.

- **No solutions**. If there are no numbers (x_1, x_2, x_3) that simultaneously satisfy all three equations, the system of equations has no solution. An example of a system of equations with no solution is the pair $s + t = 4$ and $s + t = 44$. There are no numbers (s, t) that satisfy both of these equations.

  A system of equations has no solution if its reduced row echelon form contains a row of zero coefficients with a nonzero constant in the right-hand side:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & c_1 \\ 0 & 1 & 0 & c_2 \\ 0 & 0 & 0 & c_3 \end{array}\right].$$

  If $c_3 \neq 0$, this system of equations is impossible to satisfy. There is no solution because there are no numbers (x_1, x_2, x_3) such that $0x_1 + 0x_2 + 0x_3 = c_3$.

Dear reader, we've reached the first moment in this book where you'll need to update your math vocabulary. The solution to an individual equation is a finite set of points. The *solution* to a system of equations can be an entire space containing infinitely many points, such as a line or a plane. Please update the definition of the term *solution* to include the new, more specific term *solution set*—the set of points

that satisfy the system of equations. The *solution set* of a system of three linear equations in three unknowns could be either the empty set $\{\emptyset\}$ (no solution), a set with one element $\{(x_1, x_2, x_3)\}$, or a set with infinitely many elements like a line $\{p_o + t\,\vec{v}, t \in \mathbb{R}\}$ or a plane $\{p_o + s\,\vec{v} + t\,\vec{w},\ s, t \in \mathbb{R}\}$. Another possible solution set is all of $\mathbb{R}^3$; every vector $\vec{x} \in \mathbb{R}^3$ is a solution to the equation:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Note the distinction between the three types of infinite solution sets. A line is one-dimensional, a plane is two-dimensional, and $\mathbb{R}^3$ is three-dimensional. Describing all points on a line requires one parameter, describing all points on a plane takes two parameters, and—of course—describing a point in $\mathbb{R}^3$ takes three parameters.

Geometric interpretation

We can gain some intuition about solution sets by studying the geometry of the intersections of lines in $\mathbb{R}^2$ and planes in $\mathbb{R}^3$.

Lines in two dimensions

Equations of the form $ax + by = c$ correspond to lines in $\mathbb{R}^2$. Solving systems of equations of the form

$$\begin{aligned} a_1 x + b_1 y &= c_1 \\ a_2 x + b_2 y &= c_2 \end{aligned}$$

requires finding the point $(x, y) \in \mathbb{R}^2$ where these lines intersect. There are three possibilities for the solution set:

- **One solution** if the two lines intersect at a point.
- **Infinitely many solutions** if the lines are superimposed.
- **No solution** if the two lines are parallel and never intersect.

Planes in three dimensions

Equations of the form $ax + by + cz = d$ correspond to planes in $\mathbb{R}^3$. When solving three such equations,

$$\begin{aligned} a_1 x + b_1 y + c_1 z &= d_1, \\ a_2 x + b_2 y + c_2 z &= d_2, \\ a_3 x + b_3 y + c_3 z &= d_3, \end{aligned}$$

we want to find a set of points (x, y, z) that satisfy all three equations simultaneously. There are four possibilities for the solution set:

- **One solution**. Three non-parallel planes intersect at a point.
- **Infinitely many solutions 1**. If one of the plane equations is redundant, the solution corresponds to the intersection of two planes. Two non-parallel planes intersect on a line.
- **Infinitely many solutions 2**. If two of the equations are redundant, then the solution space is a two-dimensional plane.
- **No solution**. If two (or more) of the planes are parallel, they will never intersect.

Computer power

The computer algebra system at `live.sympy.org` can be used to compute the reduced row echelon form of any matrix.

Here is an example of how to create a sympy `Matrix` object:

```
>>> from sympy.matrices import Matrix
>>> A = Matrix([[1, 2,  5],      # use SHIFT+ENTER for newline
                [3, 9, 21]])
```

In `Python`, we define lists using the square brackets `[` and `]`. A matrix is defined as a list of lists.

To compute the reduced row echelon form of `A`, call its `rref()` method:

```
>>> A.rref()
( [1, 0, 1] # RREF of A         # locations of pivots
  [0, 1, 2],                    [0, 1]               )
```

The `rref()` method returns a tuple containing the RREF of `A` and an array that tells us the 0-based indices of the columns that contain leading ones. Usually, we'll want to find the RREF of `A` and ignore the pivots; to obtain the RREF without the pivots, select the first (index zero) element in the result of `A.rref()`:

```
>>> Arref = A.rref()[0]
>>> Arref
[1, 0, 1]
[0, 1, 2]
```

Using the `rref()` method is the fastest way to obtain the reduced row echelon form of a `SymPy` matrix. The computer will apply the Gauss–Jordan elimination procedure for you and show you the answer. If you want to see the intermediary steps of the elimination procedure, you can also manually apply row operations to the matrix.

Example Let's compute the reduced row echelon form of the same augmented matrix by using row operations:

```
>>> A = Matrix([[1, 2,  5],
                [3, 9, 21]])
>>> A[1,:] = A[1,:] - 3*A[0,:]
>>> A
[1, 2, 5]
[0, 3, 6]
```

The notation `A[i,:]` is used to refer to entire rows of the matrix. The first argument specifies the 0-based row index: the first row of `A` is `A[0,:]` and the second row is `A[1,:]`. The code example above implements the row operation $R_2 \leftarrow R_2 - 3R_1$.

To obtain the reduced row echelon form of the matrix `A`, we carry out two more row operations, $R_2 \leftarrow \frac{1}{3}R_2$ and $R_1 \leftarrow R_1 - 2R_2$, using the following commands:

```
>>> A[1,:] = S(1)/3*A[1,:]
>>> A[0,:] = A[0,:] - 2*A[1,:]
>>> A
[1, 0, 1]           # the same result as A.rref()[0]
[0, 1, 2]
```

Note we represented the fraction $\frac{1}{3}$ as `S(1)/3` to obtain the exact rational expression `Rational(1,3)`. If we were to input $\frac{1}{3}$ as `1/3`, Python will interpret this either as integer or floating point division, which is not what we want. The single-letter helper function `S` is an alias for the function `sympify`, which ensures a `SymPy` object is produced. Another way to input the exact fraction $\frac{1}{3}$ is `S('1/3')`.

In case you need to swap two rows of a matrix, you can use the standard Python tuple assignment syntax. To swap the position of the first and second rows, use

```
>>> A[0,:], A[1,:] = A[1,:], A[0,:]
>>> A
[0, 1, 2]
[1, 0, 1]
```

Using row operations to compute the reduced row echelon form of a matrix allows you want to see the intermediary steps of a calculation; for instance, when checking the correctness of your homework problems.

There are other applications of matrix methods that use row operations (see Section 8.6), so it's good idea to know how to use `SymPy` for this purpose.

Discussion

In this section, we learned the Gauss–Jordan elimination procedure for simplifying matrices, which just so happens to be one of the most important computational tools of linear algebra. Beyond being a procedure for finding solutions to systems of linear equations, the Gauss–Jordan elimination algorithms can be used to solve a broad range of other linear algebra problems. Later in the book, we'll use the Gauss–Jordan elimination algorithm to compute inverse matrices (Section 4.5) and to "distill" bases for vector spaces (Section 5.5).

Exercises

E4.1 Consider the systems of equations below and its augmented matrix representation:

$$\begin{array}{rcrcl} 3x & + & 3y & = & 6 \\ 2x & + & \frac{3}{2}y & = & 5 \end{array} \quad \Rightarrow \quad \left[\begin{array}{cc|c} 3 & 3 & 6 \\ 2 & \frac{3}{2} & 5 \end{array}\right].$$

Find the solution to this system of equations by bringing the augmented matrix into reduced row echelon form.

E4.2 Repeat **E4.1** using the calculator at `http://live.sympy.org`. First define the augmented matrix using

```
>>> A = Matrix([
        [3,       3,  6],
        [2,  S(3)/2,  5]])  # note use of S(3)/2 to obtain 3/2
```

Perform row operations using `SymPy` to bring the matrix into RREF. Confirm your answer using the direct method `A.rref()`.

E4.3 Find the solutions to the systems of equations that correspond to the following augmented matrices:

a) $\left[\begin{array}{cc|c} 3 & 3 & 6 \\ 1 & 1 & 5 \end{array}\right]$ **b)** $\left[\begin{array}{cc|c} 3 & 3 & 6 \\ 2 & \frac{3}{2} & 3 \end{array}\right]$ **c)** $\left[\begin{array}{cc|c} 3 & 3 & 6 \\ 1 & 1 & 2 \end{array}\right]$

Hint: The third system of equations has a lot of solutions.

In this section we learned a practical computational algorithm for solving systems of equations by a using row operations on an augmented matrix. In the next section, we'll increase the level of abstraction: by "zooming out" one level, we can view the entire system of equations as a matrix equation $A\vec{x} = \vec{b}$ and solve the problem in one step: $\vec{x} = A^{-1}\vec{b}$.

4.2 Matrix equations

We can express the problem of solving a system of linear equations as a matrix equation and obtain the solution using the matrix inverse. Consider the following system of linear equations:

$$x_1 + 2x_2 = 5$$
$$3x_2 + 9x_2 = 21.$$

We can rewrite this system of equations using the matrix-vector product:

$$\begin{bmatrix} 1 & 2 \\ 3 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 21 \end{bmatrix},$$

or, more compactly, as

$$A\vec{x} \;=\; \vec{b},$$

where A is a 2×2 matrix, $\vec{x}$ is the vector of unknowns (a 2×1 matrix), and $\vec{b}$ is a vector of constants (a 2×1 matrix).

We can solve for $\vec{x}$ in this matrix equation by multiplying both sides of the equation by the inverse A^{-1}:

$$A^{-1}A\vec{x} = \mathbb{1}\vec{x} = \vec{x} = A^{-1}\vec{b}.$$

Thus, to solve a system of linear equations, we can find the inverse of the matrix of coefficients, then compute the product:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = A^{-1}\vec{b} = \begin{bmatrix} 3 & -\frac{2}{3} \\ -1 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 5 \\ 21 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

The computational cost of finding A^{-1} is roughly equivalent to the computational cost of bringing an augmented matrix $[\, A \mid \vec{b} \,]$ to reduced row echelon form—it's not like we're given the solution for free by simply rewriting the system of equations in matrix form. Nevertheless, expressing the system of equations as $A\vec{x} = \vec{b}$ and its solution as $\vec{x} = A^{-1}\vec{b}$ is a useful level of abstraction that saves us from needing to juggle dozens of individual coefficients. The same symbolic expression $\vec{x} = A^{-1}\vec{b}$ applies whether A is a 2×2 matrix or a 1000×1000 matrix.

Introduction

It's time we had an important discussion about matrix equations and how they differ from regular equations with numbers. If a, b, and c are three numbers, and I tell you to *solve* for a in the equation

$$ab = c,$$

you'd know the answer is $a = c/b = c\frac{1}{b} = \frac{1}{b}c$, and that would be the end of it.

Now suppose A, B, and C are matrices and you want to solve for A in the matrix equation

$$AB = C.$$

The answer $A = C/B$ is not allowed. So far, we defined matrix *multiplication* and matrix *inversion*, but not matrix *division*. Instead of dividing by B, we must multiply by B^{-1}, which, in effect, plays the same role as a "divide by B" operation. The product of B and B^{-1} gives the identity matrix,

$$BB^{-1} = \mathbb{1}, \qquad B^{-1}B = \mathbb{1}.$$

When applying the inverse matrix B^{-1} to the equation, we must specify whether we are multiplying from the left or from the right, because matrix multiplication is not commutative. Can you determine the correct answer for A in the above equations? Is it $A = CB^{-1}$ or $A = B^{-1}C$?

To solve matrix equations, we employ the same technique we used to solve equations in Chapter 1: undoing the operations that stand in the way of the unknown. Recall that we must always **do the same thing to both sides of an equation** for it to remain true.

With matrix equations, it's the same story all over again, but there are two new things you need to keep in mind:

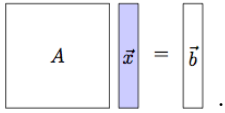
- The order in which matrices are multiplied matters because matrix multiplication is not a commutative operation $AB \neq BA$. The expressions ABC and BAC are different despite the fact that they are the product of the same three matrices.
- When performing operations on matrix equations, you can act either *from the left* or *from the right* on the equation.

The best way to familiarize yourself with the peculiarities of matrix equations is to look at example calculations. Don't worry, there won't be anything too mathematically demanding in this section; we'll just look at some pictures.

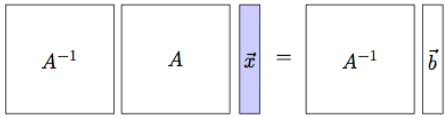
Matrix times vector

Suppose we want to solve the equation $A\vec{x} = \vec{b}$, in which an $n \times n$ matrix A multiplies the vector $\vec{x}$ to produce a vector $\vec{b}$. Recall, we can think of vectors as "tall and skinny" $n \times 1$ matrices.

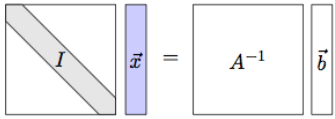
The picture corresponding to the equation $A\vec{x} = \vec{b}$ is



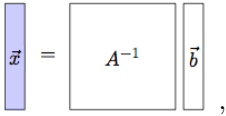
Assuming A is invertible, we can multiply by the inverse A^{-1} on the left of both sides of the equation:



By definition, A^{-1} times its inverse A is equal to the identity matrix $\mathbb{1}$, which is a diagonal matrix with ones on the diagonal and zeros everywhere else:



Any vector times the identity matrix remains unchanged, so

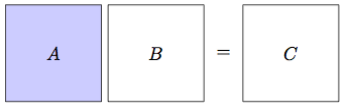


which is the final answer.

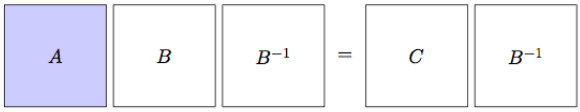
Note that the question "Solve for $\vec{x}$ in $A\vec{x} = \vec{b}$" sometimes arises in situations where the matrix A is not invertible. If the system of equations is under-specified (A is wider than it is tall), there will be a whole subspace of acceptable solutions $\vec{x}$. Recall the cases with infinite solutions (lines and planes) we saw in the previous section.

Matrix times matrix

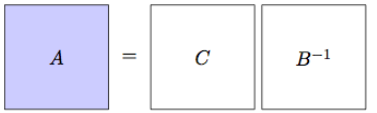
Let's look at some other matrix equations. Suppose we want to solve for A in the equation $AB = C$:



To isolate A, we multiply by B^{-1} *from the right* on both sides:

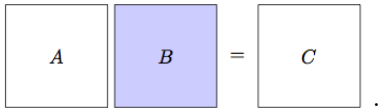


When B^{-1} hits B they cancel ($BB^{-1} = \mathbb{1}$) and we obtain the answer:

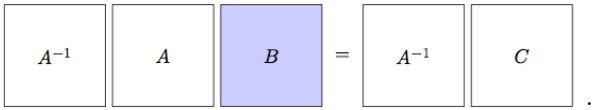


Matrix times matrix variation

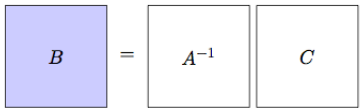
What if we want to solve for B in the same equation $AB = C$?



Again, we must *do the same* to both sides of the equation. To cancel A, we need to multiply by A^{-1} from the left:



After A^{-1} cancels with A, we obtain the final result:



This completes our lightning tour of matrix equations. There is really nothing new to learn here; just make sure you're aware that the *order* in which matrices are multiplied matters, and remember the general principle of "doing the same thing to both sides of the equation." Acting according to this principle is essential in all of math, particularly when manipulating matrices.

In the next section, we'll "zoom in" on matrix equations by examining the arithmetic operations performed on coefficients during matrix multiplication.

Exercises

E4.4 Solve for X in the following matrix equations: (1) $XA = B$; (2) $ABCXD = E$; (3) $AC = XDC$. Assume the matrices A, B, C, D, and E are invertible.

4.3 Matrix multiplication

Suppose we're given the matrices

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix},$$

and we want to compute the *matrix product* AB.

Unlike matrix addition and subtraction, matrix multiplication is *not* performed element-wise:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} \neq \begin{bmatrix} ae & bf \\ cg & dh \end{bmatrix}.$$

Instead, the matrix product is computed by taking the dot product between each row of the matrix on the left and each column of the matrix on the right:

$$\begin{array}{rcl} \vec{r}_1 \to \\ \vec{r}_2 \to \end{array} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \\ \uparrow & \uparrow \\ \vec{c}_1 & \vec{c}_2 \end{bmatrix} = \begin{bmatrix} \vec{r}_1 \cdot \vec{c}_1 & \vec{r}_1 \cdot \vec{c}_2 \\ \vec{r}_2 \cdot \vec{c}_1 & \vec{r}_2 \cdot \vec{c}_2 \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}.$$

Recall the dot product between two vectors $\vec{v}$ and $\vec{w}$ is computed using the formula $\vec{v} \cdot \vec{w} \equiv \sum_i v_i w_i$.

Now let's look at a picture that shows how to compute the product of a matrix with four rows and a matrix with five columns. To compute the top left entry, take the dot product of the first row of the matrix on the left and the first column of the matrix on the right:

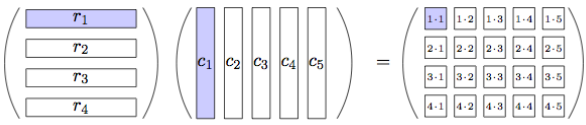


Figure 4.1: Matrix multiplication is performed "row times column." The first-row, first-column entry of the product is the dot product of r_1 and c_1.

Similarly, the entry on the third row and fourth column of the product is computed by taking the dot product of the third row of the matrix on the left and the fourth column of the matrix on the right:

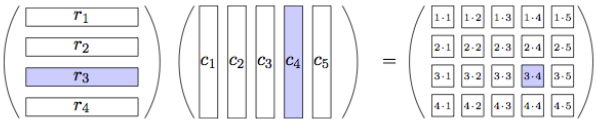


Figure 4.2: The third-row, fourth-column entry of the product is computed by taking the dot product of r_3 and c_4.

For the matrix product to work, the rows of the matrix on the left must have the same length as the columns of the matrix on the right.

Matrix multiplication rules

- Matrix multiplication is associative:

$$(AB)C = A(BC) = ABC.$$

- The "touching" dimensions of the matrices must be the same. For the triple product ABC to exist, the number of columns of A must equal to the number of rows of B, and the number of columns of B must equal the number of rows of C.
- Given two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times k}$, the product AB is an $m \times k$ matrix.
- Matrix multiplication is not a commutative operation.

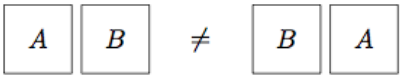


Figure 4.3: The order of multiplication matters for matrices: the product AB does not equal the product BA.

Example Consider the matrices $A \in \mathbb{R}^{2\times 3}$ and $B \in \mathbb{R}^{3\times 2}$. The product $AB = C \in \mathbb{R}^{2\times 2}$ is computed as

$$\underbrace{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}}_{B} = \begin{bmatrix}1+6+15 & 2+8+18\\ 4+15+30 & 8+20+36\end{bmatrix} = \underbrace{\begin{bmatrix}22 & 28\\ 49 & 64\end{bmatrix}}_{C}.$$

We can also compute the product $BA = D \in \mathbb{R}^{3\times 3}$:

$$\underbrace{\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}}_{B}\underbrace{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}_{A} = \begin{bmatrix}1+8 & 2+10 & 3+12\\ 3+16 & 6+20 & 9+24\\ 5+24 & 10+30 & 15+36\end{bmatrix} = \underbrace{\begin{bmatrix}9 & 12 & 15\\ 19 & 26 & 33\\ 29 & 40 & 51\end{bmatrix}}_{D}.$$

In each case, the "touching" dimensions of the two matrices in the product are the same. Note that $C = AB \neq BA = D$, and, in fact, the products AB and BA are matrices with different dimensions.

Applications

Why is matrix multiplication defined the way it is defined?

Composition of linear transformations

The long answer to this question will be covered in depth when we reach the chapter on linear transformations (Section 6, page 223). Since I don't want you to live in suspense until then, I'll give you the short answer right now. We can think of multiplying a column vector $\vec{x} \in \mathbb{R}^n$ by a matrix $A \in \mathbb{R}^{m \times n}$ as analogous to applying a "vector function" A of the form:

$$A : \mathbb{R}^n \to \mathbb{R}^m.$$

Applying the vector function A to the input $\vec{x}$ is the same as computing the matrix-vector product $A\vec{x}$:

$$\text{for all } \vec{x} \in \mathbb{R}^n, \quad A(\vec{x}) \equiv A\vec{x}.$$

Every *linear transformation* from $\mathbb{R}^n$ to $\mathbb{R}^m$ can be described as a matrix product by some matrix $A \in \mathbb{R}^{m \times n}$.

What happens when we apply two linear operations in succession? When we do this to ordinary functions, we call it *function composition*, and denote it with a little circle:

$$z = g(f(x)) = g \circ f\ (x),$$

where $g \circ f\ (x)$ indicates we should first apply f to x to obtain an intermediary value y, then apply g to y to obtain the final output z. The notation $g \circ f$ is useful when we're interested in the overall functional relationship between x and z, but don't want to talk about the intermediate result y. We can refer to the composite function $g \circ f$ and discuss its properties.

With matrices, $B \circ A$ (applying A then B) is equal to applying the product matrix BA:

$$\vec{z} = B(A(\vec{x})) = (BA)\vec{x}.$$

We can describe the overall map that transforms $\vec{x}$ to $\vec{z}$ by a single entity BA, the product of matrices B and A. **The matrix product is defined the way it is defined to enable us to easily compose linear transformations.**

Regarding matrices as linear transformations (vector functions) helps explain why matrix multiplication is not commutative. In general, $BA \neq AB$: there's no reason to expect AB will equal BA, just as there's no reason to expect that $f \circ g$ will equal $g \circ f$ for two arbitrary functions.

Matrix multiplication is an extremely useful computational tool. At the moment, your feelings about matrix multiplication might not be so warm and fuzzy, given it can be tedious and repetitive. Be patient and stick with it. Solve some exercises to make sure you understand. Afterward, you can let computers multiply matrices for you—they're good at this kind of repetitive task.

Row operations as matrix products

There is an important connection between row operations and matrix multiplication. Performing the row operation $\mathcal{R}$ on a matrix is equivalent to a left multiplication by an *elementary matrix* $E_\mathcal{R}$:

$$A' = \mathcal{R}(A) \qquad \Leftrightarrow \qquad A' = E_\mathcal{R} A.$$

There are three types of elementary matrices that correspond to the three types of row operations. Let's look at an example.

Example The row operation of adding m times the 2nd row to the 1st row ($\mathcal{R} : R_1 \leftarrow R_1 + mR_2$) corresponds to the following elementary matrix:

$$E_\mathcal{R} = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix}, \quad \text{which acts as} \quad \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a + mc & b + md \\ c & d \end{bmatrix}.$$

We'll discuss elementary matrices in more detail in Section 4.5.

We can also perform "column operations" on matrices if we multiply them by elementary matrices from the right.

Exercises

E4.5 Compute the following matrix products:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{P}, \qquad \begin{bmatrix} 3 & 1 & 2 & 2 \\ 0 & 2 & -2 & 1 \end{bmatrix} \begin{bmatrix} -2 & 3 \\ 1 & 0 \\ -2 & -2 \\ 2 & 2 \end{bmatrix} = \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{Q}.$$

4.4 Determinants

What is the volume of a rectangular box of length 1 m, width 2 m, and height 3 m? It's easy to compute the volume of this box because its shape is a *right rectangular prism*. The volume of this rectangular prism is $V = \ell \times w \times h = 6\,\text{m}^3$. What if the shape of the box was a *parallelepiped* instead? A parallelepiped is a box whose opposite faces are parallel but whose sides are slanted, as shown in Figure 4.7 on page 162. How do we compute the volume of a parallelepiped? The determinant operation, specifically the 3×3 determinant, is the perfect tool for this purpose.

The determinant operation takes square matrices as inputs and produces numbers as outputs:

$$\det : \mathbb{R}^{n \times n} \to \mathbb{R}.$$

The determinant of a matrix, denoted $\det(A)$ or $|A|$, is a particular way to multiply the entries of the matrix to produce a single number. We use determinants for all kinds of tasks: to compute areas and volumes, to solve systems of equations, to check whether a matrix is invertible or not, etc.

We can interpret the determinant of a matrix intuitively as a geometrical calculation. The determinant is the "volume" of the geometric shape whose edges are the rows of the matrix. For 2×2 matrices, the determinant corresponds to the area of a parallelogram. For 3×3 matrices, the determinant corresponds to the volume of a parallelepiped. For dimensions $d > 3$, we say the determinant measures a d-dimensional hyper-volume.

Consider the linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ defined through the matrix-vector product with a matrix A_T: $T(\vec{x}) \equiv A_T\vec{x}$. The determinant of the matrix A_T is the *scale factor* associated with the linear transformation T. The scale factor of the linear transformation T describes how the area of a unit square in the input space (a square with dimensions 1×1) is transformed by T. After passing through T, the unit square is transformed to a parallelogram with area $\det(A_T)$. Linear transformations that "shrink" areas have $\det(A_T) < 1$, while linear transformations that "enlarge" areas have $\det(A_T) > 1$. A linear transformation that is *area preserving* has $\det(A_T) = 1$.

The determinant is also used to check linear independence for a given set of vectors. We construct a matrix using the vectors as the matrix rows, and compute its determinant. If the determinant is nonzero, the vectors are linearly independent.

The determinant of a matrix tells us whether or not that matrix is invertible. If $\det(A) \neq 0$, then A is invertible; if $\det(A) = 0$, A is not invertible.

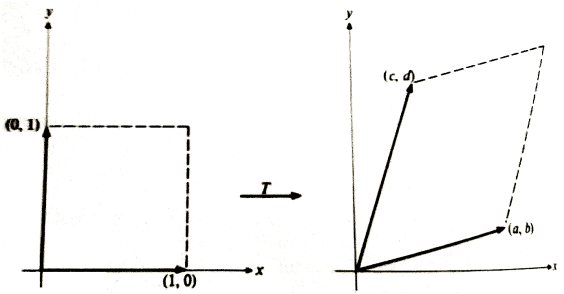


Figure 4.4: A square with side length 1 in the input space of T is transformed to a parallelogram with area $|A_T|$ in the output space of T. The determinant measures the *scale factor* by which the area changes.

The determinant shares a connection with the vector cross product, and is also used in the definition of the eigenvalue equation.

In this section, we'll discuss all the applications of determinants. As you read along, I encourage you to actively connect the geometric, algebraic, and computational aspects of determinants. Don't worry if it doesn't all click right away—you can always review this section once you've learned more about linear transformations, the geometry of cross products, and eigenvalues.

Formulas

The determinant of a 2×2 matrix is

$$\det\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\right) \equiv \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}.$$

The formulas for the determinants of larger matrices are defined recursively. For example, the determinant of a 3×3 matrix is defined in terms of 2×2 determinants:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} =$$

$$= a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

$$= a_{11}a_{22}a_{33} \quad - a_{12}a_{21}a_{33} \quad + a_{13}a_{21}a_{32}$$
$$\quad - a_{11}a_{23}a_{32} \quad + a_{12}a_{23}a_{31} \quad - a_{13}a_{22}a_{31}.$$

There's a neat computational trick for computing 3×3 determinants by hand. The trick consists of extending the matrix A into a 3×5

array that contains copies of the columns of A: the 1^{st} column of A is copied to the 4^{th} column of the extended array, and the 2^{nd} column of A is copied to the 5^{th} column. The determinant is then computed by summing the products of the entries on the three positive diagonals (solid lines) and subtracting the products of the entries on the three negative diagonals (dashed lines), as illustrated in Figure 4.5.

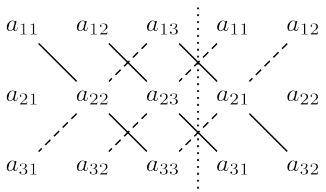


Figure 4.5: Computing the determinant using the extended-array trick.

The general formula for the determinant of an $n \times n$ matrix is

$$\det(A) = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} M_{1j},$$

where M_{ij} is called the *minor* associated with the entry a_{ij}. The minor M_{ij} is the determinant of the submatrix obtained by removing the i^{th} row and the j^{th} column of the matrix A. Note the "alternating" factor $(-1)^{i+j}$ that changes value between $+1$ and -1 for different terms in the formula.

In the case of 3×3 matrices, applying the determinant formula gives the expected expression,

$$\begin{aligned}\det(A) &= (+1)a_{11}M_{11} + (-1)a_{12}M_{12} + (+1)a_{13}M_{13} \\ &= a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}.\end{aligned}$$

The determinant of a 4×4 matrix B is

$$\det(B) = b_{11}M_{11} - b_{12}M_{12} + b_{13}M_{13} - b_{14}M_{14}.$$

The general formula for determinants $\det(A) = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} M_{1j}$, assumes we're *expanding* the determinant along the first row of the matrix. In fact, a determinant formula can be obtained by expanding the determinant along *any* row or column of the matrix. For example, expanding the determinant of a 3×3 matrix along the second column produces the determinant formula

$$\det(A) = \sum_{i=1}^{3} (-1)^{i+2} a_{i2} M_{i2} = (-1)a_{12}M_{12} + (1)a_{22}M_{22} + (-1)a_{32}M_{32}.$$

The expand-along-any-row-or-column nature of determinants can be very handy: if you need to calculate the determinant of a matrix with one row (or column) containing many zero entries, it makes sense to expand along that row since many of the terms in the formula will be zero. If a matrix contains a row (or column) consisting entirely of zeros, we can immediately tell its determinant is zero.

Geometrical interpretation

Area of a parallelogram

Suppose we're given vectors $\vec{v} = (v_1, v_2)$ and $\vec{w} = (w_1, w_2)$ in $\mathbb{R}^2$ and we construct a parallelogram with corner points $(0, 0)$, $\vec{v}$, $\vec{w}$, and $\vec{v}+\vec{w}$.

The area of this parallelogram is equal to the determinant of the matrix that contains (v_1, v_2) and (w_1, w_2) as rows:

$$\text{area} = \begin{vmatrix} v_1 & v_2 \\ w_1 & w_2 \end{vmatrix} = v_1 w_2 - v_2 w_1.$$

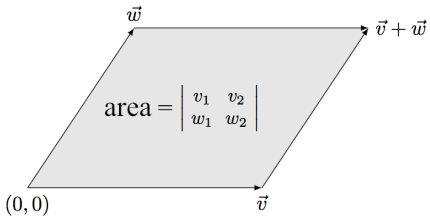


Figure 4.6: The determinant of a 2×2 matrix corresponds to the area of the parallelogram constructed from the rows of the matrix.

Volume of a parallelepiped

Suppose we are given three vectors—$\vec{u} = (u_1, u_2, u_3)$, $\vec{v} = (v_1, v_2, v_3)$, and $\vec{w} = (w_1, w_2, w_3)$ in $\mathbb{R}^3$—and we construct the parallelepiped with corner points $(0, 0, 0)$, $\vec{u}$, $\vec{v}$, $\vec{w}$, $\vec{v}+\vec{w}$, $\vec{u}+\vec{w}$, $\vec{u}+\vec{v}$, and $\vec{u}+\vec{v}+\vec{w}$, as illustrated in Figure 4.7.

The volume of this parallelepiped is equal to the determinant of the matrix containing the vectors $\vec{u}$, $\vec{v}$, and $\vec{w}$ as rows:

$$\begin{aligned} \text{volume} &= \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \\ &= u_1(v_2 w_3 - v_3 w_2) - u_2(v_1 w_3 - v_3 w_1) + u_3(v_1 w_2 - v_2 w_1). \end{aligned}$$

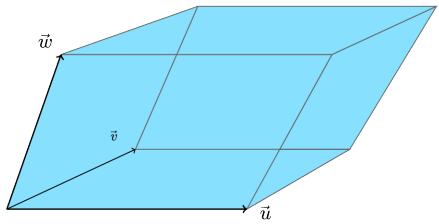


Figure 4.7: The determinant of a 3×3 matrix corresponds to the volume of the parallelepiped constructed from the rows of the matrix.

Sign and absolute value of the determinant

Calculating determinants can produce positive or negative numbers. Consider the vectors $\vec{v} = (v_1, v_2) \in \mathbb{R}^2$ and $\vec{w} = (w_1, w_2) \in \mathbb{R}^2$ and the determinant

$$D \equiv \det\left(\begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix}\right) = v_1 w_2 - v_2 w_1.$$

Let's denote the value of the determinant by the variable D. The absolute value of the determinant is equal to the area of the parallelogram constructed by the vectors $\vec{v}$ and $\vec{w}$. The sign of the determinant (positive, negative, or zero) tells us information about the relative orientation of the vectors $\vec{v}$ and $\vec{w}$. If we let θ be the measure of the angle from $\vec{v}$ toward $\vec{w}$, then

- if θ is between 0 and π[rad] (180[°]), the determinant will be positive $D > 0$. This is the case illustrated in Figure 4.6.
- if θ is between π (180[°]) and 2π[rad] (360[°]), the determinant will be negative $D < 0$.
- when $\theta = 0$ (the vectors point in the same direction), or when $\theta = \pi$ (the vectors point in opposite directions), the determinant will be zero, $D = 0$.

The formula for the area of a parallelogram is $A = b \times h$, where b is the length of the parallelogram's base, and h is the parallelogram's height. In the case of the parallelogram in Figure 4.6, the length of the base is $\|\vec{v}\|$ and the height is $\|\vec{w}\| \sin\theta$, where θ is the angle measure between $\vec{v}$ and $\vec{w}$. The geometrical interpretation of the 2×2 determinant is described by the formula,

$$D \equiv \det\left(\begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix}\right) \equiv \underbrace{\|\vec{v}\|}_{b} \underbrace{\|\vec{w}\| \sin\theta}_{h}.$$

Observe the "height" of the parallelogram is negative when θ is between π and 2π.

Properties

Let A and B be two square matrices of the same dimension. The determinant operation has the following properties:

- $\det(AB) = \det(A)\det(B) = \det(B)\det(A) = \det(BA)$
- If $\det(A) \neq 0$, the matrix is invertible and $\det(A^{-1}) = \frac{1}{\det(A)}$
- $\det\left(A^{\mathsf{T}}\right) = \det(A)$
- $\det(\alpha A) = \alpha^n \det(A)$, for an $n \times n$ matrix A
- $\det(A) = \prod_{i=1}^{n} \lambda_i$, where $\{\lambda_i\} = \text{eig}(A)$ are the eigenvalues of A

The effects of row operations on determinants

Recall the three row operations we used for the Gauss–Jordan elimination procedure:

- Add a multiple of one row to another row
- Swap two rows
- Multiply a row by a constant

We'll now describe the effects of these row operations on the value of the matrix determinant. In each case, we'll connect the effects of the row operation to the geometrical interpretation of the determinant operation.

Add a multiple of one row to another row

Adding a multiple of one row of a matrix to another row does not change the determinant of the matrix.

Figure 4.8: Row operations of the form $\mathcal{R}_\alpha : R_i \leftarrow R_i + \alpha R_j$ do not change the value of the matrix determinant.

This property follows from the fact that parallelepipeds with equal base enclosed between two parallel planes have the same volume even if they have different slants. This is known as *Cavalieri's principle*.

It is easier to visualize Cavalieri's principle in two dimensions by considering two parallelograms with base b and different slants, enclosed between two parallel lines. The area of both parallelograms is the same $A = b \times h$, where h is the distance between the parallel lines.

Swap rows

Swapping two rows of a matrix changes the sign of its determinant.

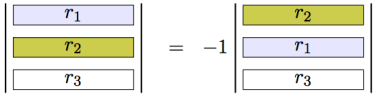


Figure 4.9: Row-swaps, $\mathcal{R}_\beta : R_i \leftrightarrow R_j$, flip the sign of the determinant.

This property is a consequence of measuring *signed volumes*. Swapping two rows changes the relative orientation of the vectors, and hence the sign of the volume.

Multiply a row by a constant

Multiplying a row by a constant is equivalent to the constant multiplying the determinant.

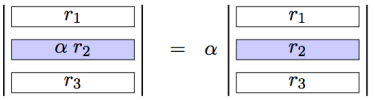


Figure 4.10: Row operations of the form $\mathcal{R}_\gamma : R_i \leftarrow \alpha R_i$ scale the value of the determinant by the factor α.

The third property follows from the fact that making one side of the parallelepiped α times longer increases the volume of the parallelepiped by a factor of α.

When each entry of an $n \times n$ matrix is multiplied by the constant α, each of the n rows is multiplied by α so the determinant changes by a factor of α^n: $\det(\alpha A) = \alpha^n \det(A)$.

Zero-vs-nonzero determinant property

There is an important distinction between matrices with zero determinant and matrices with nonzero determinant. We can understand this distinction geometrically by considering the 3×3 determinant calculation. Recall, the volume of the parallelepiped with sides $\vec{u}$, $\vec{v}$, and $\vec{w}$ is equal to the determinant of the matrix containing the vectors $\vec{u}$, $\vec{v}$, and $\vec{w}$ as rows. If the determinant is zero, it means at least one of the rows of the matrix is a linear combination of the other rows. The volume associated with this determinant is zero because the geometrical shape it corresponds to is a flattened, two-dimensional parallelepiped, in other words, a parallelogram. We say the matrix is "deficient" if its determinant is zero.

On the other hand, if the determinant of a matrix is nonzero, the rows of the matrix are linearly independent. In this case, the determinant calculation corresponds to the volume of a real parallelepiped. We say the matrix is "full" if its determinant is nonzero.

The zero-vs-nonzero determinant property of a matrix does not change when we perform row operations on the matrix. If a matrix A has a nonzero determinant, we know its reduced row echelon form will also have nonzero determinant. The number of nonzero rows in the reduced row echelon form of the matrix is called the *rank* of the matrix. We say a matrix $A \in \mathbb{R}^{n \times n}$ has *full rank* if its RREF contains n pivots. If the RREF of the matrix A contains a row of zeros, then A is not full rank and $\det(A) = 0$. On the other hand, if $\det(A) \neq 0$, we know that $\text{rref}(A) = \mathbb{1}$.

Applications

Apart from the geometric and invertibility-testing applications of determinants described above, determinants are related to many other topics in linear algebra. We'll briefly cover some of these below.

Cross product as a determinant

We can compute the cross product of the vectors $\vec{v} = (v_1, v_2, v_3)$ and $\vec{w} = (w_1, w_2, w_3)$ by computing the determinant of a special matrix. We place the symbols $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ in the first row of the matrix, then write the coefficients of $\vec{v}$ and $\vec{w}$ in the second and third rows. After expanding the determinant along the first row, we obtain the cross product:

$$
\begin{aligned}
\vec{v} \times \vec{w} &= \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \\
&= \hat{\imath} \begin{vmatrix} v_2 & v_3 \\ w_2 & w_3 \end{vmatrix} - \hat{\jmath} \begin{vmatrix} v_1 & v_3 \\ w_1 & w_3 \end{vmatrix} + \hat{k} \begin{vmatrix} v_1 & v_2 \\ w_1 & w_2 \end{vmatrix} \\
&= (v_2 w_3 - v_3 w_2)\hat{\imath} - (v_1 w_3 - v_3 w_1)\hat{\jmath} + (v_1 w_2 - v_2 w_1)\hat{k} \\
&= (v_2 w_3 - v_3 w_2,\ v_3 w_1 - v_1 w_3,\ v_1 w_2 - v_2 w_1).
\end{aligned}
$$

Observe that the anti-linear property of the vector cross product $\vec{v} \times \vec{w} = -\vec{w} \times \vec{v}$ corresponds to the swapping-rows-changes-the-sign property of determinants.

The extended-array trick for computing 3×3 determinants (see Figure 4.5) doubles as a trick for computing cross-products:

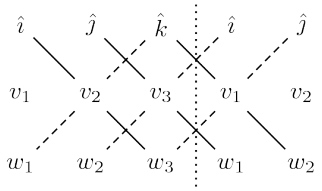


Figure 4.11: We can quickly compute the cross product of two vectors using the extended-array trick.

Using the correspondence between the cross-product and the determinant, we can write the determinant of a 3×3 matrix in terms of the dot product and cross product:

$$\begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} = \vec{u} \cdot (\vec{v} \times \vec{w}).$$

Cramer's rule

Cramer's rule is an approach for solving systems of linear equations using determinants. Consider the following system of equations and its representation as a matrix equation:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \qquad \Leftrightarrow \qquad A\vec{x} = \vec{b}.$$

We're looking for the vector $\vec{x} = (x_1, x_2, x_3)$ that satisfies this system of equations.

Begin by writing the system of equations as an augmented matrix:

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right] \equiv \left[\begin{array}{ccc|c} | & | & | & | \\ \vec{a}_1 & \vec{a}_2 & \vec{a}_3 & \vec{b} \\ | & | & | & | \end{array}\right].$$

We use the notation $\vec{a}_j$ to denote the j^{th} column of coefficients in the matrix A, and $\vec{b}$ to denote the column of constants.

Cramer's rule requires computing ratios of determinants. To find x_1, the first component of the solution vector $\vec{x}$, we compute the following

ratio of determinants:

$$x_1 = \frac{\begin{vmatrix} | & | & | \\ \vec{b} & \vec{a}_2 & \vec{a}_3 \\ | & | & | \end{vmatrix}}{\begin{vmatrix} | & | & | \\ \vec{a}_1 & \vec{a}_2 & \vec{a}_3 \\ | & | & | \end{vmatrix}} = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}.$$

Basically, we replace the column that corresponds to the unknown we want to solve for (in this case the first column) with the vector of constants $\vec{b}$, and compute the determinant before dividing by $|A|$. To find x_2, we'll need to compute the determinant of a matrix where $\vec{b}$ replaces the second column of A. Similarly, to find x_3, we replace the third column with $\vec{b}$.

Cramer's rule is a neat computational trick that might come in handy if you ever want to solve for one particular coefficient in the unknown vector $\vec{x}$, without solving for the other coefficients.

Linear independence test

Suppose you're given a set of n, n-dimensional vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ and asked to check whether the vectors are linearly independent.

You could use the Gauss–Jordan elimination procedure to accomplish this task. Write the vectors $\vec{v}_i$ as the rows of a matrix M. Next, use row operations to find the reduced row echelon form (RREF) of the matrix M. Row operations do not change the linear independence between the rows of a matrix, so you can tell whether the rows are independent from the reduced row echelon form of the matrix M.

Alternatively, you can use the *determinant test* as a shortcut to check whether the vectors are linearly independent. If $\det(M)$ is zero, the vectors that form the rows of M are not linearly independent. On the other hand, if $\det(M) \neq 0$, then the rows of M and linearly independent.

Eigenvalues

The determinant operation is used to define the *characteristic polynomial* of a matrix. The characteristic polynomial of A is

$$\begin{aligned} p_A(\lambda) &\equiv \det(A - \lambda \mathbb{1}) \\ &= \begin{vmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{vmatrix} \\ &= (a_{11} - \lambda)(a_{22} - \lambda) - a_{12}a_{21} \\ &= \lambda^2 - \underbrace{(a_{11} + a_{22})}_{\mathrm{Tr}(A)}\lambda + \underbrace{(a_{11}a_{22} - a_{12}a_{21})}_{\det(A)}. \end{aligned}$$

The roots of the characteristic polynomial are the *eigenvalues* of the matrix A. Observe the coefficient of the linear term in $p_A(\lambda)$ is equal to $-\mathrm{Tr}(A)$ and the constant term equals $\det(A)$. The name *characteristic polynomial* is indeed appropriate since $p_A(\lambda)$ encodes the information about three important properties of the matrix A: its eigenvalues (λ_1, λ_2), its trace $\mathrm{Tr}(A)$, and its determinant $\det(A)$.

At this point, we don't need to delve into a detailed discussion about properties of the characteristic polynomial. We gave the definition of $p_A(\lambda)$ here because it involves the determinant, and we're in the section on determinants. Specifically, $p_A(\lambda)$ is defined as the determinant of A with λs (the Greek letter *lambda*) subtracted from the entries on the diagonal of A. We'll continue the discussion on the characteristic polynomial and eigenvalues in Section 7.1.

Exercises

E4.6 Find the determinant of the following matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad B = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 1 & 3 & 4 \end{bmatrix}.$$

Observe that the matrix B can be obtained from the matrix A by swapping the first and second rows of the matrix. We therefore expect $\det(A)$ and $\det(B)$ to have the same absolute value but opposite signs.

E4.7 Find the volume of the parallelepiped whose sides are the vectors $\vec{u} = (1, 2, 3)$, $\vec{v} = (2, -2, 4)$, and $\vec{w} = (2, 2, 5)$.

Links

[More information about determinants from Wikipedia]
http://en.wikipedia.org/wiki/Determinant
http://en.wikipedia.org/wiki/Minor_(linear_algebra)

4.5 Matrix inverse

In this section, we'll learn four different approaches for computing the inverse of a matrix. Since knowing how to compute matrix inverses is a pretty useful skill, learning several approaches is hardly overkill. Note that the matrix inverse is *unique*, so no matter which method you use to find the inverse, you'll always obtain the same answer. You can verify your calculations by computing the inverse in different ways and checking that the answers agree.

Existence of an inverse

Not all matrices are invertible. Given a matrix $A \in \mathbb{R}^{n \times n}$, we can check whether it is invertible or not by computing its determinant:

$$A^{-1} \text{ exists} \qquad \text{if and only if} \qquad \det(A) \neq 0.$$

Calculating the determinant of a matrix serves as an *invertibility test*. The exact value of the determinant is not important; it could be big or small, positive or negative; so long as the determinant is nonzero, the matrix passes the invertibility test.

Using the adjugate matrix approach

The inverse of a 2×2 matrix can be computed as follows:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

This is the 2×2 version of a general formula for obtaining the inverse based on the *adjugate matrix*:

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A).$$

What is the adjugate matrix, you say? Ah, I'm glad you asked! The adjugate matrix is kind of complicated, so let's proceed step by step. We'll first define a few prerequisite concepts.

In the following example, we'll work on a matrix $A \in \mathbb{R}^{3\times 3}$ and refer to its entries as a_{ij}, where i is the row index and j is the column index:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

We now define three concepts associated with determinants:

1. For each entry a_{ij}, the *minor* M_{ij} is the determinant of the matrix that remains after we remove the i^{th} row and the j^{th} column of A. For example, the minor that corresponds to the entry a_{12} is given by:

$$M_{12} \equiv \begin{vmatrix} \times & \times & \times \\ a_{21} & \times & a_{23} \\ a_{31} & \times & a_{33} \end{vmatrix} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}a_{33} - a_{23}a_{31}.$$

2. The *sign* of each entry a_{ij} is defined as $\text{sign}(a_{ij}) \equiv (-1)^{i+j}$. For example, the signs of the different entries in a 3×3 matrix are

$$\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}.$$

3. The *cofactor* c_{ij} for the entry a_{ij} is the product of this entry's sign and its minor:

$$c_{ij} \equiv \text{sign}(a_{ij})M_{ij} = (-1)^{i+j}M_{ij}.$$

The above concepts should look somewhat familiar, since they previously appeared in the formula for computing determinants. If we expand the determinant along the first row of the matrix, we obtain the formula

$$\det(A) = \sum_{j=1}^{n} a_{1j}\text{sign}(a_{1j})M_{1j} = \sum_{j=1}^{n} a_{1j}c_{1j}.$$

Now you can see where the name *cofactor* comes from: the cofactor c_{ij} is what multiplies the *factor* a_{ij} in the determinant formula.

Okay, now we're ready to describe the adjugate matrix. The adjugate matrix is defined as the transpose of the *matrix of cofactors* C. The matrix of cofactors is a matrix of the same dimensions as the original matrix A that is constructed by replacing each entry a_{ij} by

its cofactor c_{ij}. The matrix of cofactors for A is

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ +\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}.$$

The adjugate matrix $\text{adj}(A)$ is the transpose of the matrix of cofactors:

$$\text{adj}(A) \equiv C^{\mathsf{T}}.$$

The formula for the inverse matrix is $A^{-1} = \frac{1}{\det(A)}\text{adj}(A)$. In the 3×3 case, the matrix inverse formula is

$$A^{-1} = \frac{1}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} \begin{bmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}.$$

I know this looks complicated, but I wanted to show you the adjugate matrix approach for computing the inverse because I think it's a nice formula. In practice, you'll rarely have to compute inverses using this approach; nevertheless, the adjugate matrix formula represents an important theoretical concept. Note the formula fails if $|A| = 0$ due to a divide-by-zero error.

Using row operations

Another way to obtain the inverse of a matrix is to record all the *row operations* $\mathcal{R}_1, \mathcal{R}_2, \ldots$ needed to transform A into the identity matrix:

$$\mathcal{R}_k(\ldots \mathcal{R}_2(\mathcal{R}_1(A))\ldots) = \mathbb{1}.$$

Recall that we can think of the action of the matrix A as a vector transformation: $\vec{w} = A\vec{v}$. By definition, the inverse A^{-1} is the operation that undoes the effect of A: $A^{-1}\vec{w} = \vec{v}$. The combination of A followed by A^{-1} is the identity transformation: $A^{-1}A\vec{v} = \mathbb{1}\vec{v} \equiv \vec{v}$.

The cumulative effect of the row operations required to transform A to the identity matrix is equivalent to the "undo" action of A.

Applying the sequence of row operations $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$ has the same effect as multiplying by A^{-1}:

$$A^{-1}\vec{w} \equiv \mathcal{R}_k(\ldots \mathcal{R}_2(\mathcal{R}_1(\vec{w}))\ldots).$$

If you think this method of finding the inverse A^{-1} seems more complicated than useful, you'd be right—if it weren't for the existence of a neat procedure for recording the row operations $\mathcal{R}_1$, $\mathcal{R}_2$,...,$\mathcal{R}_k$ that makes everything simpler. We'll discuss this procedure next.

Begin by initializing an $n \times 2n$ array with the entries of the matrix A on the left side and the identity matrix on the right side: $[\, A \mid \mathbb{1} \,]$. If we perform the Gauss–Jordan elimination procedure on this array, we'll end up with the inverse A^{-1} on the right-hand side of the array:

$$[\, A \mid \mathbb{1} \,] \; - \text{G-J elimination} \rightarrow \; [\, \mathbb{1} \mid A^{-1} \,].$$

Example Let's illustrate the procedure by computing the inverse of the following matrix:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 9 \end{bmatrix}.$$

We start by writing the matrix A next to the identity matrix $\mathbb{1}$:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 9 & 0 & 1 \end{array}\right].$$

Next, we perform the Gauss–Jordan elimination procedure on the resulting 2×4 matrix:

1. The first step is to subtract three times the first row from the second row, written compactly as $\mathcal{R}_1 : R_2 \leftarrow R_2 - 3R_1$, to obtain
$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 3 & -3 & 1 \end{array}\right].$$
2. We perform a second row operation $\mathcal{R}_2 : R_2 \leftarrow \frac{1}{3}R_2$ to obtain a leading one in the second column:
$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & -1 & \frac{1}{3} \end{array}\right].$$
3. Finally, we perform $\mathcal{R}_3 : R_1 \leftarrow R_1 - 2R_2$ to clear the entry above the leading one in the second column:
$$\left[\begin{array}{cc|cc} 1 & 0 & 3 & -\frac{2}{3} \\ 0 & 1 & -1 & \frac{1}{3} \end{array}\right].$$

The inverse of A now appears in the right-hand side of the array:

$$A^{-1} = \begin{bmatrix} 3 & -\frac{2}{3} \\ -1 & \frac{1}{3} \end{bmatrix}.$$

This algorithm works because we identify the sequence of row operations $\mathcal{R}_3(\mathcal{R}_2(\mathcal{R}_1(\ .\ )))$ with the action of the inverse matrix A^{-1}:

$$\mathcal{R}_3(\mathcal{R}_2(\mathcal{R}_1(A))) = \mathbb{1} \qquad \Rightarrow \qquad \mathcal{R}_3(\mathcal{R}_2(\mathcal{R}_1(\mathbb{1}))) = A^{-1}.$$

The combined effect of the three row operations is to "undo" the action of A, and therefore this sequence of row operations has the same effect as the inverse operation A^{-1}. The right side of the 2×4 array serves as a record of the cumulative effect of this sequence of row operations performed. Because the right side starts from a "blank" identity matrix, it will contain A^{-1} by the end of the procedure.

Using elementary matrices

Every row operation $\mathcal{R}$ performed on a matrix is equivalent to an operation of left multiplication by an *elementary matrix* $E_{\mathcal{R}}$:

$$A' = \mathcal{R}(A) \qquad \Leftrightarrow \qquad A' = E_{\mathcal{R}} A.$$

There are three types of elementary matrices that correspond to the three types of row operations. We'll illustrate the three types of elementary matrices with examples from the 2×2 case:

- Adding m times the 2^{nd} row to the 1^{st} row corresponds to

$$\mathcal{R}_\alpha : R_1 \leftarrow R_1 + mR_2 \qquad \Leftrightarrow \qquad E_\alpha = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix}.$$

- Swapping the 1^{st} and 2^{nd} rows corresponds to

$$\mathcal{R}_\beta : R_1 \leftrightarrow R_2 \qquad \Leftrightarrow \qquad E_\beta = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

- Multiplying the 1^{st} row by the constant k corresponds to

$$\mathcal{R}_\gamma : R_1 \leftarrow kR_1 \qquad \Leftrightarrow \qquad E_\gamma = \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}.$$

The general rule is simple: to find the elementary matrix that corresponds to a given row operation, apply that row operation to the identity matrix $\mathbb{1}$.

Recall the procedure we used to find the inverse in the previous section. We applied the sequence of row operations $\mathcal{R}_1, \mathcal{R}_2, \ldots$ to transform the array $[\ A \mid \mathbb{1}\ ]$ into the reduced row echelon form:

$$\mathcal{R}_k(\ldots \mathcal{R}_2(\mathcal{R}_1([\ A \mid \mathbb{1}\ ]))\ldots) \quad = \quad [\ \mathbb{1} \mid A^{-1}\ ].$$

If we represent each row operation as a multiplication by an elementary matrix, we obtain the equation

$$E_k \cdots E_2 E_1 [\ A \mid \mathbb{1}\ ] \quad = \quad [\ \mathbb{1} \mid A^{-1}\ ].$$

Observe that $E_k \cdots E_2 E_1 A = \mathbb{1}$, so we have obtained an expression for the inverse matrix A^{-1} as a product of elementary matrices:

$$A^{-1} = E_k \cdots E_2 E_1.$$

We'll now illustrate how the formula $A^{-1} = E_k \cdots E_2 E_1$ applies in the case of the matrix A discussed above

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 9 \end{bmatrix}.$$

Recall the row operations we applied in order to transform $[\ A \mid \mathbb{1}\ ]$ into $[\ \mathbb{1} \mid A^{-1}\ ]$:

1. $\mathcal{R}_1$: $R_2 \leftarrow R_2 - 3R_1$
2. $\mathcal{R}_2$: $R_2 \leftarrow \frac{1}{3}R_2$
3. $\mathcal{R}_3$: $R_1 \leftarrow R_1 - 2R_2$

Let's revisit these row operations, representing each of them as a multiplication by an elementary matrix:

1. The first row operation $\mathcal{R}_1 : R_2 \leftarrow R_2 - 3R_1$ corresponds to a multiplication by the elementary matrix E_1:

$$E_1 = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}, \qquad E_1 A = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}.$$

2. The second row operation $\mathcal{R}_2 : R_2 \leftarrow \frac{1}{3}R_2$ corresponds to a matrix E_2:

$$E_2 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{bmatrix}, \qquad E_2(E_1 A) = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$

3. The third row operation $\mathcal{R}_3 : R_1 \leftarrow R_1 - 2R_2$ corresponds to the elementary matrix E_3:

$$E_3 = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}, \quad E_3(E_2 E_1 A) = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that $E_3E_2E_1A = \mathbb{1}$, so the product $E_3E_2E_1$ must equal A^{-1}:

$$A^{-1} = E_3E_2E_1 = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -\frac{2}{3} \\ -1 & \frac{1}{3} \end{bmatrix}.$$

Verify the last equation by computing the product of the three elementary matrices.

Since we know $A^{-1} = E_3E_2E_1$, then $A = (A^{-1})^{-1} = (E_3E_2E_1)^{-1} = E_1^{-1}E_2^{-1}E_3^{-1}$. We can write A as a product of elementary matrices:

$$A = E_1^{-1}E_2^{-1}E_3^{-1} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$

The inverses of the elementary matrices are trivial to compute; they correspond to elementary "undo" operations.

The elementary matrix approach teaches us that every invertible matrix A can be decomposed as the product of elementary matrices. The inverse matrix A^{-1} consists of the product of the inverses of the elementary matrices that make up A (in the reverse order).

Using a computer algebra system

You can use a computer algebra system to specify matrices and compute their inverses. Let's illustrate how to find the matrix inverse using the computer algebra system at `live.sympy.org`.

```
>>> from sympy.matrices import Matrix
>>> A = Matrix( [ [1,2],[3,9] ] )   # define a Matrix object
>>> A.inv()                         # call the inv method on A
[ 3, -2/3]
[-1,  1/3]
```

Note `SymPy` returns an answer in terms of exact rational numbers. This is in contrast with numerical computer algebra systems like `Octave` and `MATLAB`, which are based on floating point arithmetic.

You can use `SymPy` to check your answers on homework problems.

Discussion

We've explored several ways to compute matrix inverses. If you need to find the inverse of a matrix using only pen and paper, on a final exam for example, I recommend using the Gauss–Jordan elimination procedure on the extended array:

$$[\ A \mid \mathbb{1}\ ] \ - \text{G–J elimination} \rightarrow \ [\ \mathbb{1} \mid A^{-1}\ ],$$

since it is fairly easy to perform even for large matrices and it leverages your experience with the Gauss–Jordan elimination procedure.

For 2×2 matrices, the formula $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ is faster.

Invertibility

Not all matrices are invertible. Keep this in mind, since teachers might try to trick you by asking you to find the inverse of a noninvertible matrix. Let's analyze how each procedure for computing the inverse fails when applied to a noninvertible matrix D. The inverse formula based on the determinant and the adjugate matrix is $D^{-1} = \frac{1}{\det(D)}\text{adj}(D)$. However, if the matrix D is not invertible, then $\det(D) = 0$ and the formula fails due to a divide-by-zero error. The row operations approach to computing the inverse will also fail. Starting from the extended array $[\ D \mid \mathbb{1}\ ]$, you can apply all the row operations you want, but you'll never be able to obtain the identity matrix in the left-hand side of the array. This is because the reduced row echelon form of a noninvertible matrix D has at least one row of zeros: $\text{rref}(D) \neq \mathbb{1}$. We'll discuss invertible matrices and their properties in Section 6.4. For now, be sure to remember the *determinant test* for invertibility: if $\det(A) = 0$, then A is noninvertible, and if $\det(A) \neq 0$, then A is invertible.

Exercises

E4.8 Compute A^{-1} where $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$.

E4.9 Implement the matrix inverse formula $A^{-1} = \frac{1}{\det(A)}\text{adj}(A)$ for the case of 3×3 using a spreadsheet.

E4.10 Show that for an $n \times n$ invertible matrix A, the determinant of the adjugate matrix is $|\text{adj}(A)| = (|A|)^{n-1}$.

Hint: Recall that $|A^{-1}| = \frac{1}{|A|}$ and $|\alpha A| = \alpha^n |A|$.

4.6 Computational problems

P4.1 Mitchell wants to eat healthily. His target is to eat exactly 25 grams of fat and 32 grams of protein for lunch today. There are two types of food in the fridge, x and y. One serving of food x contains one gram of fat and two grams of protein, while a serving of food y contains five grams of fat and one gram of protein. To figure out how many servings of each type of food he should eat, he comes up with the following system of equations:

$$\begin{array}{rcrcl} x & + & 5y & = & 25 \\ 2x & + & y & = & 32 \end{array} \quad \Rightarrow \quad \left[\begin{array}{cc|c} 1 & 5 & 25 \\ 2 & 1 & 32 \end{array}\right].$$

Can you help Mitchell find how many servings of x and y he should eat?

Hint: Find the reduced row echelon form of the augmented matrix.

P4.2 Alice, Bob, and Charlotte are solving this systems of equations:

$$\begin{array}{rcl} 3x & +3y & = 6 \\ 2x & +\frac{3}{2}y & = 5 \end{array} \quad \Rightarrow \quad \left[\begin{array}{cc|c} 3 & 3 & 6 \\ 2 & \frac{3}{2} & 5 \end{array}\right].$$

Alice follows the "standard" procedure to obtain a leading one, by performing the row operation $R_1 \leftarrow \frac{1}{3}R_1$. Bob starts with a different row operation, applying $R_1 \leftarrow R_1 - R_2$ to obtain a leading one. Charlotte takes a third approach by swapping the first and second rows: $R_1 \leftrightarrow R_2$.

a) $\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 2 & \frac{3}{2} & 5 \end{array}\right]$ **b)** $\left[\begin{array}{cc|c} 1 & \frac{3}{2} & 1 \\ 2 & \frac{3}{2} & 5 \end{array}\right]$ **c)** $\left[\begin{array}{cc|c} 2 & \frac{3}{2} & 5 \\ 3 & 3 & 6 \end{array}\right]$

Help Alice, Bob, and Charlotte finish the problems by writing the list of remaining row operations each of them must perform to bring their version of the augmented matrix into reduced row echelon form.

P4.3 Find the solutions to the systems of equations that correspond to the following augmented matrices:

a) $\left[\begin{array}{cc|c} -1 & -2 & -2 \\ 3 & 3 & 0 \end{array}\right]$ **b)** $\left[\begin{array}{ccc|c} 1 & -1 & -2 & 1 \\ -2 & 3 & 3 & -1 \\ -1 & 0 & 1 & 2 \end{array}\right]$ **c)** $\left[\begin{array}{ccc|c} 2 & -2 & 3 & 2 \\ 1 & -2 & -1 & 0 \\ -2 & 2 & 2 & 1 \end{array}\right]$

P4.4 Find the solution set for the systems of equations described by the following augmented matrices:

a) $\left[\begin{array}{cc|c} -1 & -2 & -2 \\ 3 & 6 & 6 \end{array}\right]$ **b)** $\left[\begin{array}{ccc|c} 1 & -1 & -2 & 1 \\ -2 & 3 & 3 & -1 \\ -1 & 2 & 1 & 0 \end{array}\right]$ **c)** $\left[\begin{array}{ccc|c} 2 & -2 & 3 & 2 \\ 0 & 0 & 5 & 3 \\ -2 & 2 & 2 & 1 \end{array}\right]$

P4.5 Find the solution to the following systems of equations:

a) $\left[\begin{array}{ccc|c} 2 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 4 & 2 & -2 & 0 \end{array}\right]$ **b)** $\left[\begin{array}{ccc|c} 2 & 0 & 1 & 5 \\ 1 & 4 & 2 & 2 \\ 0 & 2 & 1 & 1 \end{array}\right]$ **c)** $\left[\begin{array}{ccc|c} 2 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 4 & 2 & -2 & 0 \end{array}\right]$

P4.6 Solve for C in the matrix equation $ABCD = AD$.

P4.7 Solve for the following matrix equations problems:

(a) Simplify the expression $MNB^{-1}BK^{-1}KN^{-1}M^{-2}L^{-1}S^{-1}SMK^2$.
(b) Simplify $J^{-3}K^2G^{-1}GK^{-3}J^2$.
(c) Solve for A in the equation $A^{-1}BNK = B^2B^{-1}NK$.
(d) Solve for Y in $SUNNY = SUN$.

You can assume all matrices are invertible.

P4.8 Solve for $\vec{x}$ in $A\vec{x} = \vec{b}$, where $A = \left[\begin{smallmatrix} 1 & 0 & -3 \\ 2 & -1 & 1 \\ 0 & 0 & -1 \end{smallmatrix}\right]$ and $\vec{b} = (2, 2, 3)^{\mathsf{T}}$.

P4.9 Solve for $\vec{x}$ in the equation $\vec{x} = \vec{d} + A\vec{x}$, where $A = \left[\begin{smallmatrix} 0 & 0.05 & 0.3 \\ 0.01 & 0 & 0.01 \\ 0.1 & 0 & 0 \end{smallmatrix}\right]$, and $\vec{d} = (25, 10, 14)^{\mathsf{T}}$. Use `live.sympy.org` to perform the calculations.

Hint: Rewrite as $\mathbb{1}\vec{x} = \vec{d} + A\vec{x}$ then bring all the $\vec{x}$s to one side.

P4.10 Given the following two matrices,

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 4 & 2 \\ 3 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 3 & 1 \\ 2 & 0 \\ 1 & 1 \end{bmatrix},$$

compute the matrix products a) AB, b) AA, c) BA, and d) BB.

P4.11 Compute the following product of three matrices:

$$\begin{bmatrix} 2 & 10 & -5 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 0 & 2 \\ 5 & 1 \\ -3 & -4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

P4.12 Given an unknown variable $\alpha \in \mathbb{R}$ and the matrices

$$A = \begin{bmatrix} \cos(\alpha) & 1 \\ -1 & -\sin(\alpha) \end{bmatrix}; \quad B = \begin{bmatrix} \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) \end{bmatrix}; \quad C = \begin{bmatrix} 1 & -\cos(\alpha) \\ \sin(\alpha) & 1 \end{bmatrix},$$

compute the value of a) $A^2 + B^2$, b) $A^2 + C$, c) $A^2 + C - B^2$. Give you answer in terms of α and use double-angle formulas as needed.

P4.13 Find the determinants of the following matrices:

a) $\begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 0 & 5 & 3 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 2 & 0 \\ 3 & 1 & 1 \\ 4 & -2 & 0 \end{bmatrix}$

P4.14 Find the determinants of these matrices

$$A = \begin{bmatrix} 3 & -1 & 5 & 2 \\ 0 & 2 & 2 & -3 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}; \qquad B = \begin{bmatrix} 2 & -1 & 0 & -3 & 2 \\ 0 & 1 & 1 & 3 & 0 \\ 1 & 4 & 0 & 0 & -1 \\ 3 & -2 & 3 & 1 & 0 \\ -1 & 0 & -1 & 0 & 2 \end{bmatrix}.$$

P4.15 Find and area of a parallelogram which has vectors $\vec{v} = (3, -5)$ and $\vec{w} = (1, -1)$ as its sides.

Hint: Use the formula from Section 4.4 (page 161).

P4.16 Find volume of the parallelepiped who has the vectors $\vec{u} = (2, 0, 1)$, $\vec{v} = (1, -1, 1)$ and $\vec{w} = (0, 2, 3)$ as sides.

P4.17 Given the matrix

$$A = \begin{bmatrix} 3 & -2 & 0 & 1 \\ 0 & 1 & 3 & -1 \\ 5 & 0 & 1 & 4 \\ 0 & 3 & -4 & 2 \end{bmatrix},$$

a) Find the determinant of A.
b) Find the determinant when you interchange the 1^{st} and 3^{rd} rows.
c) Find the determinant after multiplying the 2^{nd} row by -2.

P4.18 Check whether the rows of the following matrices are linearly independent or not:

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}; \qquad B = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 1 & 1 \\ 0 & -2 & -1 \end{bmatrix};$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & -2 & 2 & -2 \\ -4 & 4 & -4 & 4 \\ -8 & 8 & -8 & 8 \end{bmatrix}; \qquad D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & -2 & 0 \\ -1 & 2 & 1 & -2 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Are the *columns* of these matrices linearly independent?

P4.19 The transformation from *polar coordinates* (r, θ) to cartesian coordinates (x, y) is given the equations $x(r, \theta) = r\cos\theta$ and $y(r, \theta) = r\sin\theta$. Under this transformation area changes as $dxdy = \det(J)drd\theta$, where $\det(J)$ is the *area scaling factor* of the transformation. The matrix J contains the partial derivates of $x(r, \theta)$ and $y(r, \theta)$, and is called the Jacobian matrix of the transformation:

$$J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix}.$$

Compute the value of $\det(J)$.

P4.20 Spherical coordinates (ρ, θ, ϕ) are described by

$$\begin{aligned} x &= \rho\sin\phi\cos\theta, \\ y &= \rho\sin\phi\sin\theta, \\ z &= \rho\cos\phi. \end{aligned}$$

Small "volume chunks" transforms according to $dxdydz = \det(J_s)d\phi d\theta d\rho$, where $\det(J_s)$ is the *volume scaling factor*, computed as the determinant of the Jacobian matrix:

$$J_s = \begin{bmatrix} \frac{\partial x}{\partial \rho} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial \rho} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \phi} \\ \frac{\partial z}{\partial \rho} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \phi} \end{bmatrix}.$$

Compute the absolute value of $\det(J_s)$.

P4.21 Find the inverses for the following matrices:

a) $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$ c) $\begin{bmatrix} 2 & 3 \\ 2 & 4 \end{bmatrix}$

P4.22 Given the matrix equation $AB = C$, where A and C are 2×2 matrices, find the matrix B.

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 7 \end{bmatrix} \qquad C = \begin{bmatrix} 3 & 2 \\ 1 & -4 \end{bmatrix}.$$

P4.23 Find an inverse of the following matrix:

$$A = \begin{bmatrix} 0 & -3 & 2 & 4 \\ 1 & -1 & 1 & -1 \\ 2 & 4 & 0 & -2 \\ 3 & 0 & 1 & 0 \end{bmatrix}.$$

P4.24 Prove that the zero matrix A has no inverse.

P4.25 Obtain the matrices of cofactors for the following matrices:

$$A=\begin{bmatrix}1 & 4 & 3\\ 2 & 1 & 1\\ 0 & -2 & -1\end{bmatrix}; \qquad B=\begin{bmatrix}5 & 0 & 1\\ 3 & -1 & -3\\ 0 & -4 & -2\end{bmatrix}; \qquad C=\begin{bmatrix}1 & 1 & 1 & 1\\ 2 & 0 & -2 & 0\\ -1 & 2 & 1 & -2\\ 1 & -1 & -1 & 1\end{bmatrix}.$$

P4.26 Find a, b, c and d in this equation:

$$\begin{bmatrix}1 & 3\\ -2 & -1\end{bmatrix}\begin{bmatrix}a & b\\ c & d\end{bmatrix}=\begin{bmatrix}3 & -5\\ 4 & 0\end{bmatrix}.$$

P4.27 Given the constraints $a = g$, $e = b = f$, and $c = d = h$, find a choice of the variables a, b, c, d, e, f, g, h that satisfy the matrix equation:

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix}\begin{bmatrix}e & f\\ g & h\end{bmatrix}=\begin{bmatrix}-2 & -3\\ 0 & 2\end{bmatrix}.$$

Chapter 5

Geometrical aspects of linear algebra

In this section we'll study geometrical objects like lines, planes, and vector spaces. We'll use what we learned about vectors and matrices in the previous chapters to perform geometrical calculations such as projections and distance measurements.

Developing your intuition about the geometrical problems of linear algebra is very important: of all the things you learn in this course, your geometrical intuition will stay with you the longest. Years from now, you may not recall the details of the Gauss–Jordan elimination procedure, but you'll still remember that the solution to three linear equations in three variables corresponds to the intersection of three planes in $\mathbb{R}^3$.

5.1 Lines and planes

Points, *lines*, and *planes* are the basic building blocks of geometry. In this section, we'll explore these geometric objects, the equations that describe them, and their visual representations.

Concepts

- $p = (p_x, p_y, p_z)$: a *point* in $\mathbb{R}^3$
- $\vec{v} = (v_x, v_y, v_z)$: a *vector* in $\mathbb{R}^3$
- $\hat{v} = \frac{\vec{v}}{\|\vec{v}\|}$: the *unit vector* in the same direction as the vector $\vec{v}$
- An infinite line ℓ is a one-dimensional space defined in one of several possible ways:

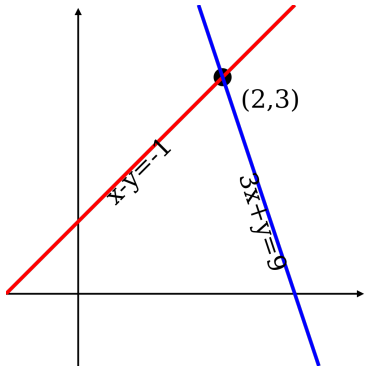
- ▷ $\ell : \{p_o + t\,\vec{v}, t \in \mathbb{R}\}$: a *parametric equation* of a line with direction vector $\vec{v}$ passing through the point p_o
- ▷ $\ell : \left\{ \frac{x-p_{ox}}{v_x} = \frac{y-p_{oy}}{v_y} = \frac{z-p_{oz}}{v_z} \right\}$: a *symmetric equation*

- An infinite plane P is a two-dimensional space defined in one of several possible ways:
  - ▷ $P : \{Ax + By + Cz = D\}$: a *general equation*
  - ▷ $P : \{p_o + s\,\vec{v} + t\,\vec{w},\ s,t \in \mathbb{R}\}$: a *parametric equation*
  - ▷ $P : \{\vec{n} \cdot [(x,y,z) - p_o] = 0\}$: a *geometric equation* of the plane that contains point p_o and has normal vector $\hat{n}$
- $d(a,b)$: the shortest *distance* between geometric objects a and b

Points

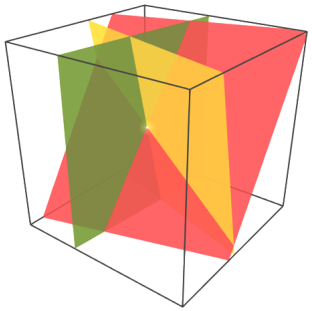
We can specify a point in $\mathbb{R}^3$ by its coordinates $p = (p_x, p_y, p_z)$, which is similar to how we specify vectors. In fact, the two notions are equivalent: we can either talk about the destination point p or the vector $\vec{p}$ that takes us from the origin to the point p. This equivalence lets us add and subtract vectors and points. For example, $\vec{d} = q - p$ denotes the displacement vector that takes the point p to the point q.

We can also specify a point as the intersection of two lines. As an example in $\mathbb{R}^2$, let's define $p = (p_x, p_y)$ to be the intersection of the lines $x - y = -1$ and $3x + y = 9$. We must solve the two equations simultaneously to find the coordinates of the point p. We can use the standard techniques for solving equations to find the answer. The intersection point is $p = (2,3)$. Note that for two lines to intersect at a point, the lines must not be parallel.



Example 1 Find where the lines $x + 2y = 5$ and $3x + 9y = 21$ intersect. To find the point of intersection, we solve these equations simultaneously and obtain the point (x, y) that is contained in both lines. The answer is the point $p = (1, 2)$.

In three dimensions, a point can also be specified as the intersection of three planes. This is precisely what happens when we solve equations of the form:



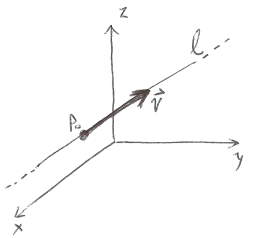
$$A_1x + B_1y + C_1z = D_1,$$
$$A_2x + B_2y + C_2z = D_2,$$
$$A_3x + B_3y + C_3z = D_3.$$

To solve this system of equations, we must find the point (x, y, z) that satisfies all three equations, which means this point is contained in all three planes.

Lines

A line ℓ is a one-dimensional space that is infinitely long. There are several equivalent ways to specify a line in space.

The *parametric equation* of a line is obtained as follows. Given a direction vector $\vec{v}$ and some point p_o on the line, we define the line as the following set:



$$\ell : \{(x, y, z) \in \mathbb{R}^3 \mid (x, y, z) = p_o + t\vec{v}, t \in \mathbb{R}\}.$$

The line consists of all the points (x, y, z) that can be reached starting from the point p_o and adding any multiple of the direction vector $\vec{v}$. We say the line is *parametrized* by the variable t.

The *symmetric equation* is an equivalent way to describe a line that does not require an explicit parametrization. Consider the equations that correspond to each coordinate in the parametric equation of a line:

$$x = p_{ox} + t\,v_x, \quad y = p_{oy} + t\,v_y, \quad z = p_{oz} + t\,v_z.$$

When we solve for t in these equations and equate the results, we obtain the *symmetric equation* of a line:

$$\ell : \left\{ \frac{x - p_{ox}}{v_x} = \frac{y - p_{oy}}{v_y} = \frac{z - p_{oz}}{v_z} \right\}.$$

Note the parameter t does not appear. The symmetric equation specifies the line as the relationships between the x, y, and z coordinates that hold for all points on the line.

You're probably most familiar with the symmetric equation of lines in $\mathbb{R}^2$, which do not involve the variable z. For non-vertical lines

in $\mathbb{R}^2$ ($v_x \neq 0$), we can think of y as a function of x and write the equation of the line in the equivalent form:

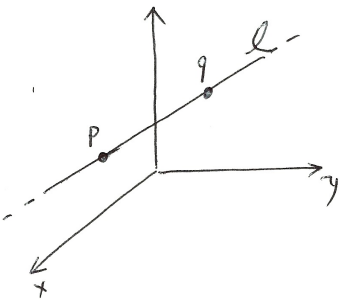
$$\frac{x - p_{ox}}{v_x} = \frac{y - p_{oy}}{v_y} \qquad \Rightarrow \qquad y(x) = mx + b,$$

where $m = \frac{v_y}{v_x}$ and $b = p_{oy} - \frac{v_y}{v_x} p_{ox}$. The equation $m = \frac{v_y}{v_x}$ makes sense intuitively: the slope of a line m corresponds to how much the line "moves" in the y-direction divided by how much the line "moves" in the x-direction.

Another way to describe a line is to specify two points that are part of the line. The equation of a line that contains the points p and q can be obtained as follows:

$$\ell : \ \{\vec{x} = p + t\,(q - p), \ t \in \mathbb{R}\},$$

where $(q - p)$ plays the role of the direction vector $\vec{v}$ for this line. Any vector parallel to the line can be used as the direction vector for the line.

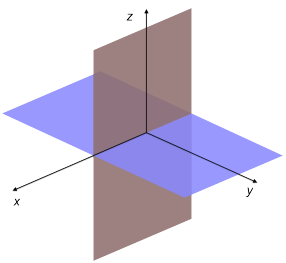


Example 2 Find the parametric equation of the line that passes through the points $p = (1, 1, 1)$ and $q = (2, 3, 4)$. What is the symmetric equation of this line?

Using the direction vector $\vec{v} = q - p = (1, 2, 3)$ and the point p on the line, we can write a parametric equation for the line as $\{(x, y, z) \in \mathbb{R}^3 \mid (x, y, z) = (1, 1, 1) + t(1, 2, 3), t \in \mathbb{R}\}$. Note that a parametric equation using the direction vector $(-1, -2, -3)$ would be equally valid: $\{(1, 1, 1) + t(-1, -2, -3), t \in \mathbb{R}\}$. The symmetric equation of the line is $\frac{x-1}{1} = \frac{y-1}{2} = \frac{z-1}{3}$.

Lines as intersections of planes

In three dimensions, the intersection of two non-parallel planes forms a line. For example, the intersection of the xy-plane P_{xy} : $\{(x, y, z) \in \mathbb{R}^3 \mid z = 0\}$ and the xz-plane $P_{xz} : \{(x, y, z) \in \mathbb{R}^3 \mid y = 0\}$ is the x-axis: $\{(x, y, z) \in \mathbb{R}^3 \mid (0, 0, 0) + (1, 0, 0)t, \ t \in \mathbb{R}\}$. For this simple case, we can imagine the two planes (use your hands) and visually establish that they intersect along the x-axis. Wouldn't it be nice if there was a general procedure for finding the line of intersection of two planes?



You already know such a procedure! The line of intersection between the planes $A_1x + B_1y + C_1z = D_1$ and $A_2x + B_2y + C_2z = D_2$ is the solution of the following set of linear equations:

$$A_1x + B_1y + C_1z = D_1,$$
$$A_2x + B_2y + C_2z = D_2.$$

Example 3 Find the intersection of the planes $0x+0y+1z = 0$ and $0x + 1y + 1z = 0$. We follow the standard Gauss–Jordan elimination procedure: construct an augmented matrix, perform row operations (denoted $\sim$), obtain the RREF, and interpret the solution:

$$\left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right].$$

The first column is a free variable $t \in \mathbb{R}$. The solution is the line

$$\left\{ \begin{array}{l} x = t \\ y = 0 \\ z = 0 \end{array}, \quad \forall t \in \mathbb{R} \right\} = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \forall t \in \mathbb{R} \right\},$$

which corresponds to the x-axis.

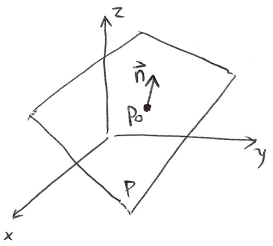
Planes

A plane P in $\mathbb{R}^3$ is a two-dimensional space with infinite extent. In general, we specify a plane through a constraint equation that must be satisfied by all points in the plane:

$$P: \ \{(x, y, z) \in \mathbb{R}^3 \mid Ax + By + Cz = D\}.$$

The plane P is the set of all points $(x, y, z) \in \mathbb{R}^3$ that satisfy the equation $Ax + By + Cz = D$. The equation $Ax + By + Cz = D$ is called the *general equation* of the plane. This definition represents the *algebraic view* of planes, which is useful for calculations.

There is an equally useful geometric view of planes. A plane can be specified by a *normal vector* $\vec{n}$ and some point p_o in the plane. The normal vector $\vec{n}$ is perpendicular to the plane: it sticks out at right angles to the plane like the normal force between surfaces in physics problems. All points in the plane P can be obtained starting from the point p_o and moving in a direction orthogonal to the normal vector $\vec{n}$. The geometric formula of a plane is



$$P: \quad \vec{n} \cdot [(x, y, z) - p_o] = 0.$$

Recall that the dot product of two vectors is zero if and only if these vectors are orthogonal. In the above equation, the expression $[(x, y, z) - p_o]$ forms an arbitrary vector with one endpoint at p_o. From all these vectors, we select *only* those that are perpendicular to $\vec{n}$ and thus we obtain all the points in the plane.

The geometric equation $\vec{n} \cdot [(x, y, z) - p_o] = 0$ is equivalent to the general equation $Ax + By + Cz = D$. We can find the parameters A, B, C, and D by calculating the dot product: $A = n_x, B = n_y, C = n_z$, and $D = \vec{n} \cdot p_o = n_x p_{ox} + n_y p_{oy} + n_y p_{oz}$.

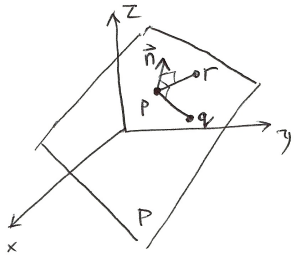
Observe that scaling the general equation of a plane by a constant factor does not change the plane: the equations $Ax + By + Cz = D$ and $\alpha Ax + \alpha By + \alpha Cz = \alpha D$ define the same plane. Similarly the geometric equations $\vec{n} \cdot [(x, y, z) - p_o] = 0$ and $\alpha\vec{n} \cdot [(x, y, z) - p_o] = 0$ define the same plane. In each case, the direction of the normal vector $\vec{n}$ is important, but not its length.

We can also give a parametric equation of a plane P. If we know a point p_o in the plane and two linearly independent vectors $\vec{v}$ and $\vec{w}$ that lie in the plane, then a parametric equation for the plane can be obtained as follows:

$$P: \ \{(x, y, z) \in \mathbb{R}^3 \mid (x, y, z) = p_o + s\,\vec{v} + t\,\vec{w}, \ s, t \in \mathbb{R}\}.$$

Since a plane is a two-dimensional space, we need two parameters (s and t) to describe the location of arbitrary points in the plane.

Suppose we're given three points p, q, and r that lie in the plane. How can we find the geometric equation for this plane $\vec{n} \cdot [(x, y, z) - p_o] = 0$? We can use the point p as the reference point p_o, but how do we find the normal vector $\vec{n}$ for the plane? The trick is to use the cross product. First we build two vectors that are parallel to the plane, $\vec{v} = q - p$ and $\vec{w} = r - p$, and then compute their cross product to find a vector that is perpendicular to both of them, and hence normal to the plane.



$$\vec{n} = \vec{v} \times \vec{w} = (q - p) \times (r - p).$$

We can use the vector $\vec{n}$ to write the geometric equation of the plane $\vec{n} \cdot [(x, y, z) - p] = 0$. The key property we used is the fact that the cross product of two vectors is perpendicular to both vectors. The cross product is the perfect tool for finding normal vectors.

Example 4 Consider the plane that contains the points $p = (1, 0, 0)$, $q = (0, 1, 0)$, and $r = (0, 0, 1)$. Find a geometric equation, a general equation, and a parametric equation for this plane.

We need a normal vector for the geometric equation. We can obtain a normal vector from the cross product of the vectors $\vec{v} = q - p = (-1, 1, 0)$ and $\vec{w} = r - p = (-1, 0, 1)$, which both lie in the plane. We obtain the normal $\vec{n} = \vec{v} \times \vec{w} = (1, 1, 1)$ and write the geometric equation as $(1, 1, 1) \cdot [(x, y, z) - (1, 0, 0)] = 0$, using p as the point p_o in the plane. To find the general equation for the plane, we compute the dot product in the geometric equation and obtain $1x + 1y + 1z - 1 = 0$, which is the same as $x + y + z = 1$. The vectors $\vec{v}$ and $\vec{w}$ obtained above can also be used in the parametric equation of the plane: $\{(1, 0, 0) + s(-1, 1, 0) + t(-1, 0, 1),\ s, t \in \mathbb{R}\}$.

Distance formulas

We'll now discuss three formulas for calculating distances: the distance between two points, the closest distance between a line and the origin, and the closest distance between a plane and the origin.

Distance between points

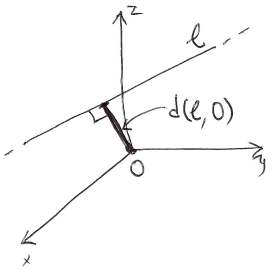
The distance between points p and q is equal to the length of the vector that goes from p to q:

$$d(p, q) = \|q - p\| = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2 + (q_z - p_z)^2}.$$

Distance between a line and the origin

The closest distance between the line with equation $\ell : \{p_o + t\,\vec{v}, t \in \mathbb{R}\}$ and the origin $O = (0, 0, 0)$ is given by the formula

$$d(\ell, O) = \left\| p_o - \frac{p_o \cdot \vec{v}}{\|\vec{v}\|^2} \vec{v} \right\|.$$



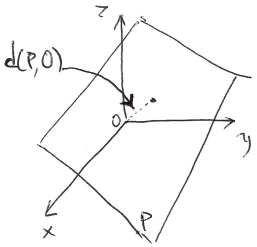
Example 5 The closest distance between the line $\ell : \{(4, 5, 6) + t(1, 0, 1),\ t \in \mathbb{R}\}$ and the origin $O = (0, 0, 0)$ is calculated as follows:

$$\begin{aligned} d(\ell, O) &= \left\| (4, 5, 6) - \frac{(4, 5, 6) \cdot (1, 0, 1)}{1^2 + 0^2 + 1^2} (1, 0, 1) \right\| \\ &= \left\| (4, 5, 6) - \frac{4 + 0 + 6}{2} (1, 0, 1) \right\| \\ &= \|(-1, 5, 1)\| \ = \ 3\sqrt{3}. \end{aligned}$$

Distance between a plane and the origin

The closest distance between the plane with geometric equation $P: \ \vec{n} \cdot [(x, y, z) - p_o] = 0$ and the origin O is given by

$$d(P, O) = \frac{|\vec{n} \cdot p_o|}{\|\vec{n}\|}.$$



For example, the distance between the plane $P: \ (-3, 0, -4) \cdot [(x, y, z) - (1, 2, 3)] = 0$ and the origin is computed as

$$d(P, O) = \frac{|(-3, 0, -4) \cdot (1, 2, 3)|}{\|(-3, 0, -4)\|} = \frac{|-3 - 12|}{5} = \frac{15}{5} \ = \ 3.$$

Discussion

The distance formulas given above are complicated expressions that involve calculating dot products and taking the length of vectors. To understand the logic behind these distance formulas, we need to learn a bit about *projective geometry*. The techniques of projective geometry allow us to measure distances between arbitrary points, lines, and planes. No new math operations are required. Instead, we'll learn how to use a combination of vector subtraction, vector length, and the dot product to compute distances. Each distance function $d(\cdot, \cdot)$ corresponds to an abstract procedure with one or two steps which can be described using a vector diagram. Projections play a key role in projective geometry, so we'll learn about them in detail in the next section.

Exercises

E5.1 Find the closest distance between $\ell : \{(4, 5, 6) + t(7, 8, 9), \ t \in \mathbb{R}\}$ and the origin.

E5.2 Find the distance between the plane P with geometric equation $(1, 1, 1) \cdot [(x, y, z) - (4, 5, 6)] = 0$ and the origin.

E5.3 Two nonparallel planes in $\mathbb{R}^3$ intersect at a line. Two intersecting nonparallel lines ℓ_1 and ℓ_2 in $\mathbb{R}^3$ define a unique plane $P : \vec{n} \cdot [(x, y, z) - p_o] = 0$ that contains both lines. More generally, a pair of nonintersecting nonparallel lines ℓ_1 and ℓ_2 in $\mathbb{R}^3$ defines a whole family of planes $\{\vec{n} \cdot (x, y, z) = d, \ \forall d \in \mathbb{R}\}$ that are parallel to both lines, that is, they never intersect with the two lines.

Find the equation of the plane that contains the line of intersection of the planes $x + 2y + z = 1$ and $2x - y - z = 2$ and is parallel to the line with parametric equation $x = 1 + 2t$, $y = -2 + t$, $z = -1 - t$.

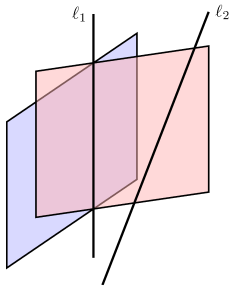


Figure 5.1: The line of intersection ℓ_1 of two planes and another line ℓ_2. In **E5.3** we want to find the plane that contains ℓ_1 and does't intersect ℓ_2.

E5.4 Find the general equation of the line that passes through the points $(0, 5)$ and $(6, -7)$ in $\mathbb{R}^2$.

5.2 Projections

In this section we'll learn to compute projections of vectors onto lines and planes. Given an arbitrary vector, you'll need to find how much of this vector points in a given direction (projection onto a line), or you'll need to find the part of the vector that lies in some plane (projection onto a plane). The dot product, $\vec{u} \cdot \vec{v} \equiv u_1v_1 + u_2v_2 + u_3v_3$, will play a central role in these calculations.

Each projection formula corresponds to a vector diagram. Vector diagrams, also known as "picture proofs," are used to describe the precise sequence of operations used to compute a projection. Focussing on the vector diagrams will make it much easier to understand the projection and distance formulas. Indeed, the pictures in this section are a heck of a lot more important than the formulas. Be sure you understand each vector diagram well and don't worry about memorizing the corresponding formula. You can easily reproduce the formula by starting from the vector diagram.

Concepts

- $S \subseteq \mathbb{R}^n$: S is a *vector subspace* of $\mathbb{R}^n$. In this chapter, we assume $S \subseteq \mathbb{R}^3$. The subspaces of $\mathbb{R}^3$ are lines ℓ and planes P that pass through the origin.
- $S^\perp$: the orthogonal space to S, $S^\perp \equiv \{\vec{w} \in \mathbb{R}^n \mid \vec{w} \cdot S = 0\}$. The symbol $^\perp$ stands for *perpendicular to*.
- Π_S: the *projection* onto the subspace S
- $\Pi_{S^\perp}$: the projection onto the orthogonal space $S^\perp$

Definitions

Let S be a *vector subspace* of $\mathbb{R}^n$, denoted $S \subseteq \mathbb{R}^n$. In this section, we'll focus on the subspaces of the space $\mathbb{R}^3$. The vector subspaces of $\mathbb{R}^3$ are lines and planes that pass through the origin.

The projection operation onto the subspace S is a linear transformation that takes as inputs vectors in $\mathbb{R}^3$, and produces outputs in the subspace S:

$$\Pi_S : \mathbb{R}^3 \ \rightarrow \ S.$$

The transformation Π_S cuts off all parts of the input that do not lie within the subspace S. We can understand Π_S by analyzing its action for different inputs:

- If $\vec{v} \in S$, then $\Pi_S(\vec{v}) = \vec{v}$.
- If $\vec{w} \in S^\perp$, then $\Pi_S(\vec{w}) = \vec{0}$.
- Linearity and the above two conditions imply that, for any vector $\vec{u} = \alpha\vec{v} + \beta\vec{w}$ with $\vec{v} \in S$ and $\vec{w} \in S^\perp$, we have

$$\Pi_S(\vec{u}) = \Pi_S(\alpha\vec{v} + \beta\vec{w}) = \alpha\vec{v}.$$

The *orthogonal subspace* to S is the set of vectors that are perpendicular to all vectors in S:

$$S^\perp \equiv \{ \, \vec{w} \in \mathbb{R}^3 \mid \vec{w} \cdot \vec{s} = 0, \ \forall \vec{s} \in S \, \}.$$

The operator Π_S "projects" to the space S in the sense that, no matter which vector $\vec{u}$ you start from, applying the projection Π_S will result in a vector that is in S:

$$\forall \vec{u} \in \mathbb{R}^3, \quad \Pi_S(\vec{u}) \in S.$$

All parts of $\vec{u}$ that were in the *perp*-space $S^\perp$ will be killed by Π_S. Meet Π_S—the *S-perp* killer.

We can split the set of all vectors $\mathbb{R}^3$ into two disjoint sets: vectors entirely contained in S and vectors perpendicular to S. We say $\mathbb{R}^3$ decomposes into the *direct sum* of the subspaces S and $S^\perp$:

$$\mathbb{R}^3 = S \oplus S^\perp.$$

Any vector $\vec{u} \in \mathbb{R}^3$ can be split into an S-part $\vec{v} = \Pi_S(\vec{u})$ and a $S^\perp$-part $\vec{w} = \Pi_{S^\perp}(\vec{u})$, such that

$$\vec{u} = \vec{v} + \vec{w}.$$

A defining property of projections is that they are *idempotent operations*, meaning it doesn't matter if you project a vector once, twice, or a million times; the result will always be the same:

$$\Pi_S(\vec{u}) = \Pi_S(\Pi_S(\vec{u})) = \Pi_S(\Pi_S(\Pi_S(\vec{u}))) = \ldots.$$

Once you project a vector onto the subspace S, any further projections to S will have no effect.

In the remainder of this section, we'll derive formulas for projections onto lines and planes that pass through the origin.

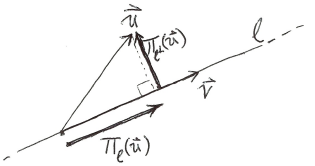
Projection onto a line

Consider the line ℓ passing through the origin with direction vector $\vec{v}$:

$$\ell: \ \{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z) = \vec{0} + t\,\vec{v}, \ t \in \mathbb{R}\}.$$

The projection onto ℓ for an arbitrary vector $\vec{u} \in \mathbb{R}^3$ is given by the formula

$$\Pi_\ell(\vec{u}) = \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|^2}\vec{v}.$$



The orthogonal space to the line ℓ consists of all vectors perpendicular to the direction vector $\vec{v}$. Mathematically speaking,

$$\ell^\perp: \ \ \{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z)\cdot\vec{v} = 0\}.$$

Recognize that the equation $(x,y,z)\cdot\vec{v} = 0$ defines a *plane*. The orthogonal space for a line ℓ with direction vector $\vec{v}$ is a plane with normal vector $\vec{v}$. Makes sense, yes?

We can easily find the projection operation onto $\ell^\perp$ as well. Any vector can be written as the sum of an ℓ part and a $\ell^\perp$ part: $\vec{u} = \vec{v}+\vec{w}$, where $\vec{v} = \Pi_\ell(\vec{u}) \in \ell$ and $\vec{w} = \Pi_{\ell^\perp}(\vec{u}) \in \ell^\perp$. To obtain $\Pi_{\ell^\perp}(\vec{u})$ we subtract the Π_ℓ part from the original vector $\vec{u}$:

$$\Pi_{\ell^\perp}(\vec{u}) = \vec{w} = \vec{u} - \vec{v} = \vec{u} - \Pi_\ell(\vec{u}) = \vec{u} - \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|^2}\vec{v}.$$

We can think of $\Pi_{\ell^\perp}(\vec{u}) = \vec{w}$ as the part of $\vec{u}$ that remains after we've removed the ℓ-part.

Example 1 Consider the line ℓ defined by the parametric equation $\{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z) = t(1,2,3), t \in \mathbb{R}\}$. Find the projection of the vector $\vec{u} = (4,5,6)$ onto ℓ. Find the projection of $\vec{u}$ onto $\ell^\perp$ and verify that $\Pi_\ell(\vec{u}) + \Pi_{\ell^\perp}(\vec{u}) = \vec{u}$.

The direction vector of the line ℓ is $\vec{v} = (1,2,3)$, so $\Pi_\ell(\vec{u}) = \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|^2}\vec{v} = \frac{16}{7}\vec{v} = (\frac{16}{7}, \frac{32}{7}, \frac{48}{7})$. Next, using the formula $\Pi_{\ell^\perp}(\vec{u}) = \vec{u} - \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|^2}\vec{v}$, we find $\Pi_{\ell^\perp}(\vec{u}) = (\frac{12}{7}, \frac{3}{7}, \frac{-6}{7})$. Observe that $(\frac{12}{7}, \frac{3}{7}, \frac{-6}{7})\cdot\vec{v} = 0$, which shows the vector $\Pi_{\ell^\perp}(\vec{u})$ is indeed perpendicular to ℓ. Adding the results of the two projections, we obtain the whole $\vec{u}$: $(\frac{16}{7}, \frac{32}{7}, \frac{48}{7}) + (\frac{12}{7}, \frac{3}{7}, \frac{-6}{7}) = (\frac{28}{7}, \frac{35}{7}, \frac{42}{7}) = (4,5,6) = \vec{u}$.

Projection onto a plane

Now consider the two-dimensional plane P passing through the origin with normal vector $\vec{n}$:

$$P: \quad \{(x,y,z) \in \mathbb{R}^3 \mid \vec{n} \cdot (x,y,z) = 0\}.$$

The perpendicular space $S^\perp$ is a line with direction vector $\vec{n}$:

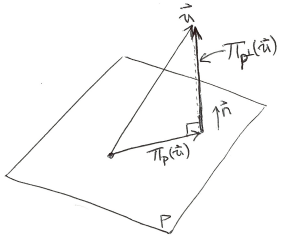
$$P^\perp: \ \{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z) = \vec{0} + t\,\vec{n}, \ t \in \mathbb{R}\}.$$

Again, the vector space $\mathbb{R}^3$ decomposes into the direct sum of P and $P^\perp$: $\mathbb{R}^3 = P \oplus P^\perp$.

We want to find Π_P, but it will actually be easier to find $\Pi_{P^\perp}$ first and then compute $\Pi_P(\vec{u})$ using $\Pi_P(\vec{u}) = \vec{v} = \vec{u} - \vec{w}$, where $\vec{w} = \Pi_{P^\perp}(\vec{u})$.

Since $P^\perp$ is a line, we know the formula for projecting onto it is

$$\Pi_{P^\perp}(\vec{u}) = \frac{\vec{u} \cdot \vec{n}}{\|\vec{n}\|^2}\vec{n}.$$



We can now obtain the formula for Π_P:

$$\Pi_P(\vec{u}) = \vec{v} = \vec{u} - \vec{w} = \vec{u} - \Pi_{P^\perp}(\vec{u}) = \vec{u} - \frac{\vec{u} \cdot \vec{n}}{\|\vec{n}\|^2}\vec{n}.$$

Example 2 Consider the plane P defined by the geometric equation $(1,1,1) \cdot [(x,y,z) - (0,0,0)] = 0$. Find the projection of the vector $\vec{u} = (4,5,6)$ onto P and onto $P^\perp$. Verify that $\Pi_P(\vec{u}) + \Pi_{P^\perp}(\vec{u}) = \vec{u}$.

Using the formula $\Pi_P(\vec{u}) = \vec{u} - \frac{\vec{u}\cdot\vec{n}}{\|\vec{n}\|^2}\vec{n}$, we find $\Pi_P(\vec{u}) = (-1,0,1)$. We also find $\Pi_{P^\perp}(\vec{u}) = \frac{\vec{u}\cdot\vec{n}}{\|\vec{n}\|^2}\vec{n} = (5,5,5)$, which is a vector in the same direction as $\vec{n}$. Observe that the vector $\vec{u}$ can be reconstructed by adding the two projections: $\Pi_P(\vec{u}) + \Pi_{P^\perp}(\vec{u}) = (-1,0,1) + (5,5,5) = (4,5,6) = \vec{u}$.

Distances formulas revisited

Suppose you want to find the distance between the line $\ell : \{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z) = p_o + t\,\vec{v}, t \in \mathbb{R}\}$ and the origin $O = (0,0,0)$. This problem is equivalent to finding the distance between the line $\ell' : \{(x,y,z) \in \mathbb{R}^3 \mid (x,y,z) = \vec{0} + t\,\vec{v}, t \in \mathbb{R}\}$ and the point p_o, the answer to which is the length of the projection $\Pi_{\ell^\perp}(p_o)$:

$$d(\ell', p_o) = \|\Pi_{\ell^\perp}(p_o)\| = \left\| p_o - \frac{p_o \cdot \vec{v}}{\|\vec{v}\|^2}\vec{v} \right\|.$$

The distance between a plane $P: \vec{n} \cdot [(x, y, z) - p_o] = 0$ and the origin O is the same as the distance between the plane $P': \vec{n} \cdot (x, y, z) = 0$ and the point p_o. We can obtain this distance by finding the length of the projection of p_o onto $P'^{\perp}$ using the formula

$$d(P', p_o) = \frac{|\vec{n} \cdot p_o|}{\|\vec{n}\|}.$$

You should try drawing the pictures for the above two scenarios and make sure the formulas make sense to you.

Projections matrices

Since projections are *linear transformations*, they can be expressed as matrix-vector products:

$$\vec{v} = \Pi(\vec{u}) \qquad \Leftrightarrow \qquad \vec{v} = M_\Pi \vec{u}.$$

Multiplying the vector $\vec{u}$ by the matrix M_Π is the same as applying the projection Π.

We'll learn more about projection matrices later. For now, I'll show you a simple example of a projection matrix in $\mathbb{R}^3$. Let Π be the projection onto the xy-plane. This projection operation corresponds to the following matrix-vector product:

$$\Pi(\vec{u}) = M_\Pi \vec{u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ 0 \end{bmatrix}.$$

Note how multiplying a vector by M_Π results in selecting only the x- and y-components of the vector while killing the z-component, which is precisely what the projection onto the xy-plane is supposed to do.

Discussion

In the next section we'll talk about a particular set of projections known as *coordinate projections*. We use coordinate projections to find the components of vectors $\vec{v}$ with respect to a coordinate system:

$$v_x\hat{\imath} \equiv \Pi_x(\vec{v}) = \frac{\vec{v} \cdot \hat{\imath}}{\|\hat{\imath}\|^2}\hat{\imath} = (\vec{v} \cdot \hat{\imath})\hat{\imath},$$

$$v_y\hat{\jmath} \equiv \Pi_y(\vec{v}) = \frac{\vec{v} \cdot \hat{\jmath}}{\|\hat{\jmath}\|^2}\hat{\jmath} = (\vec{v} \cdot \hat{\jmath})\hat{\jmath},$$

$$v_z\hat{k} \equiv \Pi_z(\vec{v}) = \frac{\vec{v} \cdot \hat{k}}{\|\hat{k}\|^2}\hat{k} = (\vec{v} \cdot \hat{k})\hat{k}.$$

The coordinate projection Π_x projects onto the x-axis, and similarly Π_y and Π_z project onto the y- and z-axes.

Exercises

E5.5 Find the orthogonal projection of the vector $\vec{v} = (4, 5, 6)$ onto the plane that contains the vectors $(2, -4, 2)$ and $(6, 1, -4)$.

5.3 Coordinate projections

In science, it's common to express vectors as components: (v_x, v_y, v_z). Thinking of vectors as arrays of numbers is fine for computational purposes, since vector operations require manipulating the components of vectors. However, this way of thinking about vectors overlooks an important concept—the *basis* with respect to which the vector's components are expressed.

It's not uncommon for students to have misconceptions about linear algebra due to an incomplete understanding of the fundamental distinction between vectors and their components. Since I want you to have a *thorough* understanding of linear algebra, we'll review—in full detail—the notion of a basis and how to compute vector components with respect to different bases.

Example In physics we learn how to work with vectors in terms of their x- and y-components. Given a standard xy-coordinate system, we can decompose a force vector $\vec{F}$ in terms of its components:

$$F_x = \|\vec{F}\| \cos\theta, \qquad F_y = \|\vec{F}\| \sin\theta,$$

where θ is the angle the vector $\vec{F}$ makes with the x-axis. We can express the vector as *coordinates* with respect to the basis $\{\hat{\imath}, \hat{\jmath}\}$ as $\vec{F} = F_x\hat{\imath} + F_y\hat{\jmath} = (F_x, F_y)_{\hat{\imath}\hat{\jmath}}$.

The number F_x corresponds to the length of the *projection* of the vector $\vec{F}$ onto the x-axis. In the last section, we discussed the projection operation and learned how to compute projections using the dot product or as a matrix-vector product:

$$F_x\,\hat{\imath} \;=\; \frac{\vec{F}\cdot\hat{\imath}}{\|\hat{\imath}\|^2}\hat{\imath} \;=\; \Pi_x(\vec{F}) \;=\; \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}_{M_{\Pi_x}} \underbrace{\begin{bmatrix} F_x \\ F_y \end{bmatrix}}_{\vec{F}},$$

where $\Pi_x : \mathbb{R}^2 \to \mathbb{R}^2$ is the projection operator onto the x-axis (a linear transformation) and M_{Π_x} is its matrix representation with respect to the basis $\{\hat{\imath}, \hat{\jmath}\}$. Now we'll discuss the relationship between vectors and their representation in terms of coordinates with respect to different bases.

Concepts

We can define three different types of bases for an n-dimensional vector space V:

- A generic basis $B_f = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n\}$ consists of any set of linearly independent vectors in V.
- An orthogonal basis $B_e = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ consists of n mutually orthogonal vectors in V obeying $\vec{e}_i \cdot \vec{e}_j = 0, \forall i \neq j$.
- An orthonormal basis $B_{\hat{e}} = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ is an orthogonal basis of unit length vectors: $\hat{e}_i \cdot \hat{e}_j = 0, \forall i \neq j$ and $\hat{e}_i \cdot \hat{e}_i = \|\hat{e}_i\| = 1, \forall i \in \{1, 2, \ldots, n\}$.

A vector $\vec{v}$ is expressed as coordinates v_i with respect to any basis B:

$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n = (v_1, v_2, \ldots, v_n)_B.$$

We can use two different bases B and B' to express the same vector:

- $\vec{v}$: a vector
- $[\vec{v}]_B = (v_1, v_2, \ldots, v_n)_B$: the vector $\vec{v}$ expressed in the basis B
- $[\vec{v}]_{B'} = (v'_1, v'_2, \ldots, v'_n)_{B'}$: the same vector $\vec{v}$ expressed in a different basis B'
- ${}_{B'}[\mathbb{1}]_B$: the change-of-basis matrix that converts from B coordinates to B' coordinates: $[\vec{v}]_{B'} = {}_{B'}[\mathbb{1}]_B \, [\vec{v}]_B$.

Components with respect to a basis

A vector's *components* describe how much of the vector lies in a given direction. For example, a vector $\vec{v} \in \mathbb{R}^3$ expressed as components *with respect to* the standard orthonormal basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ is denoted $\vec{v} = (v_x, v_y, v_z)_{\hat{\imath}\hat{\jmath}\hat{k}}$. The *components* of a vector are also called *coordinates* (in the context of a coordinate system) and *coefficients* (in the context of a linear combination). Don't be confused by this multitude of terms: it's the same idea.

When the basis consists of a set of orthonormal vectors like the vectors $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$, we can compute vector components using the dot product:

$$v_x = \vec{v} \cdot \hat{\imath}, \qquad v_y = \vec{v} \cdot \hat{\jmath}, \qquad v_z = \vec{v} \cdot \hat{k}.$$

In this section, we'll discuss how to find coordinates with respect to three different types of bases: orthonormal bases, orthogonal bases, and generic bases. First, let's give a precise definition of what a *basis* is.

Definition of a basis

A basis $B = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ for the vector space V has the following two properties:

- **Spanning property**. Any vector $\vec{v} \in V$ can be expressed as a linear combination of the basis elements:

$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n.$$

  This property guarantees that the vectors in the basis B are *sufficient* to represent any vector in V.
- **Linear independence property**. The vectors that form the basis $B = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ are linearly independent. The linear independence of the vectors in the basis guarantees that none of the vectors $\vec{e}_i$ is redundant.

If a set of vectors $B = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ satisfies both properties, we say B is a basis for V. In other words, B can serve as a coordinate system for V.

Coordinates with respect to an orthonormal basis

An orthonormal basis $B_{\hat{e}} = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ consists of a set of mutually orthogonal, unit-length vectors:

$$\hat{e}_i \cdot \hat{e}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

The vectors are mutually orthogonal since $\hat{e}_i \cdot \hat{e}_i$ for all $i \neq j$, and the vectors have unit length since $\hat{e}_i \cdot \hat{e}_i = 1$ implies $\|\hat{e}_i\| = 1$.

To compute the components of the vector $\vec{a}$ with respect to an orthonormal basis $B_{\hat{e}}$ we use the standard "prescription" similar to the one we used for the $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ basis:

$$(a_1, a_2, \ldots, a_n)_{B_{\hat{e}}} \;\Leftrightarrow\; \underbrace{(\vec{a} \cdot \hat{e}_1)}_{a_1}\hat{e}_1 \;+\; \underbrace{(\vec{a} \cdot \hat{e}_2)}_{a_2}\hat{e}_2 \;+\; \cdots \;+\; \underbrace{(\vec{a} \cdot \hat{e}_n)}_{a_n}\hat{e}_n.$$

Coordinates with respect to an orthogonal basis

Consider a basis that is orthogonal, but not orthonormal: $B_e = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$. We can compute the coordinates of any vector $\vec{b}$ with respect to the basis $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ as follows:

$$(b_1, b_2, \ldots, b_n)_{B_e} \;\Leftrightarrow\; \left(\frac{\vec{b} \cdot \vec{e}_1}{\|\vec{e}_1\|^2}\right)\vec{e}_1 + \left(\frac{\vec{b} \cdot \vec{e}_2}{\|\vec{e}_2\|^2}\right)\vec{e}_2 + \cdots + \left(\frac{\vec{b} \cdot \vec{e}_n}{\|\vec{e}_n\|^2}\right)\vec{e}_n.$$

To find the coefficients of the vector $\vec{b}$ with respect to B_e, we use the general projection formula:

$$b_1 = \frac{\vec{b} \cdot \vec{e}_1}{\|\vec{e}_1\|^2}, \quad b_2 = \frac{\vec{b} \cdot \vec{e}_2}{\|\vec{e}_2\|^2}, \quad \cdots, \quad b_n = \frac{\vec{b} \cdot \vec{e}_n}{\|\vec{e}_n\|^2}.$$

Observe that each component of the vector can be computed independently of the other components: to compute b_1, all we need to know is $\vec{b}$ and $\vec{e}_1$; we don't need to know $\vec{e}_2$, $\vec{e}_3$, ..., $\vec{e}_n$, because we know the other basis vectors are orthogonal to $\vec{e}_1$. The computation of the coefficients correspond to n independent *orthogonal projections*. The coefficient b_i tells us how much of the basis vector $\vec{e}_i$ we need in the linear combination to construct the vector $\vec{b}$.

Coordinates with respect to a generic basis

Suppose we're given a generic basis $\{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n\}$ for a vector space. To find the coefficients $(c_1, c_2, \ldots, c_n)$ of a vector $\vec{c}$ with respect to this basis, we need to solve the equation

$$c_1\vec{f}_1 + c_2\vec{f}_2 + \cdots + c_n\vec{f}_n = \vec{c}$$

for the unknowns $c_1, c_2, \ldots, c_n$. Because the vectors $\{\vec{f}_i\}$ are not orthogonal, the calculation of the coefficients $c_1, c_2, \ldots, c_n$ must be done simultaneously.

Example Express the vector $\vec{v} = (5, 6)_{\hat{\imath}\hat{\jmath}}$ in terms of the basis $B_f = \{\vec{f}_1, \vec{f}_2\}$ where $\vec{f}_1 = (1, 1)_{\hat{\imath}\hat{\jmath}}$ and $\vec{f}_2 = (3, 0)_{\hat{\imath}\hat{\jmath}}$.

We are looking for the coefficients v_1 and v_2 such that

$$(v_1, v_2)_{B_f} = v_1\vec{f}_1 + v_2\vec{f}_2 = \vec{v} = (5, 6)_{\hat{\imath}\hat{\jmath}}.$$

To find the coefficients we need to solve the following system of equations *simultaneously*:

$$\begin{aligned} 1v_1 + 3v_2 &= 5 \\ 1v_1 + 0 &= 6. \end{aligned}$$

From the second equation we find $v_1 = 6$. We substitute this answer into the first equation and find $v_2 = \frac{-1}{3}$. Thus, the vector $\vec{v}$ written with respect to the basis $\{\vec{f}_1, \vec{f}_2\}$ is $\vec{v} = 6\vec{f}_1 - \frac{1}{3}\vec{f}_2 = \left(6, \frac{-1}{3}\right)_{B_f}$.

The general case of computing a vector's coordinates with respect to a generic basis $\{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n\}$ requires solving a system of n equations in n unknowns. You know how to do this, but it will take some

work. The take-home message is that orthogonal and orthonormal bases are easiest to work with, since computing vector coordinates requires only computing projections as opposed to solving systems of equations.

Change of basis

We often identify a vector $\vec{v}$ with its components (v_x, v_y, v_z). It's important to always keep in mind the basis with respect to which the coefficients are taken, and if necessary specify the basis as a subscript $\vec{v} = (v_x, v_y, v_z)_{\hat{\imath}\hat{\jmath}\hat{k}}$. When performing vector arithmetic operations like $\vec{u} + \vec{v}$, we don't really care what basis the vectors are expressed in so long as the *same* basis is used for both $\vec{u}$ and $\vec{v}$.

We sometimes need to use multiple bases, however. Consider the basis $B = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ and another basis $B' = \{\hat{e}'_1, \hat{e}'_2, \hat{e}'_3\}$. Suppose we're given the coordinates v_1, v_2, v_3 of some vector $\vec{v}$ in terms of the basis B:

$$\vec{v} = (v_1, v_2, v_3)_B = v_1\hat{e}_1 + v_2\hat{e}_2 + v_3\hat{e}_3.$$

How can we find the coefficients of $\vec{v}$ in terms of the basis B'?

This is called a *change-of-basis* transformation and is performed as a matrix multiplication with a *change-of-basis matrix*:

$$\begin{bmatrix} v'_1 \\ v'_2 \\ v'_3 \end{bmatrix}_{B'} = \underbrace{\begin{bmatrix} \hat{e}'_1 \cdot \hat{e}_1 & \hat{e}'_1 \cdot \hat{e}_2 & \hat{e}'_1 \cdot \hat{e}_3 \\ \hat{e}'_2 \cdot \hat{e}_1 & \hat{e}'_2 \cdot \hat{e}_2 & \hat{e}'_2 \cdot \hat{e}_3 \\ \hat{e}'_3 \cdot \hat{e}_1 & \hat{e}'_3 \cdot \hat{e}_2 & \hat{e}'_3 \cdot \hat{e}_3 \end{bmatrix}}_{{}_{B'}[\mathbb{1}]_B} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_B .$$

The columns of the change-of-basis matrix describe the vectors of the basis $\{\hat{e}_i\}$ in terms of the basis $\{\hat{e}'_i\}$.

Note that multiplying a vector by the matrix ${}_{B'}[\mathbb{1}]_B$ doesn't actually *do* anything since it doesn't change the vector. The change-of-basis operation acts like the identity transformation, which is why we use the notation ${}_{B'}[\mathbb{1}]_B$ to describe it. The vector $\vec{v}$ stays the same—it is simply expressed in terms of a different basis:

$$(v'_1, v'_2, v'_3)_{B'} = v'_1\hat{e}'_1 + v'_2\hat{e}'_2 + v'_3\hat{e}'_3 = \vec{v} = v_1\hat{e}_1 + v_2\hat{e}_2 + v_3\hat{e}_3 = (v_1, v_2, v_3)_B.$$

We say the vector $\vec{v}$ has two *representations*. The vector $\vec{v}$ corresponds to the triple of coefficients (v_1, v_2, v_3) with respect to the basis B and to the triple (v'_1, v'_2, v'_3) with respect to the basis B'.

The matrix ${}_{B'}[\mathbb{1}]_B$ contains the information about how each vector of the old basis (B) is expressed in terms of the new basis (B'). For example, the first column of the change-of-basis matrix describes how the vector $\hat{e}_1$ is expressed in terms of the basis B':

$$\hat{e}_1 = (\hat{e}'_1 \cdot \hat{e}_1)\,\hat{e}'_1 + (\hat{e}'_2 \cdot \hat{e}_1)\,\hat{e}'_2 + (\hat{e}'_3 \cdot \hat{e}_1)\,\hat{e}'_3 = (\hat{e}'_1 \cdot \hat{e}_1,\ \hat{e}'_2 \cdot \hat{e}_1,\ \hat{e}'_3 \cdot \hat{e}_1)_{B'} .$$

Note this is the generic formula for expressing vectors in the basis B'.

To find the entries of the change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ between orthonormal bases B and B', it's sufficient to compute all the dot products $\hat{e}'_i \cdot \hat{e}_j$. To compute the entries of a change-of-basis matrix between bases B and B' (which are orthogonal but not necessarily orthonormal) we use $({}_{B'}[\mathbb{1}]_B)_{ij} = \frac{\vec{e}'_i \cdot \vec{e}_j}{\|\vec{e}'_i\| \, \|\vec{e}_j\|}$. Computing the change-of-basis matrix between nonorthogonal bases is more complicated.

Links

[Khan Academy video on the change-of-basis operation]
`http://youtube.com/watch?v=meibWcbGqt4`

Exercises

5.4 Vector spaces

Get ready—we're about to shift gears from individual vectors to entire sets of vectors. We're entering the territory of *vector spaces*. For instance, the set of all possible three-dimensional vectors is denoted $\mathbb{R}^3$, and is a type of *vector space*. A vector space consists of a set of vectors and all linear combinations of these vectors. This means if the vectors $\vec{v}_1$ and $\vec{v}_2$ are part of some vector space, then so is the vector $\alpha\vec{v}_1 + \beta\vec{v}_2$ for any α and β. A *vector subspace* consists of a subset of all possible vectors. The vector subspaces of $\mathbb{R}^3$ are lines and planes that pass through the origin.

Since vector spaces and subspaces play a central role in many areas of linear algebra, you'll want to learn about the properties of vector spaces and develop your vocabulary for describing them.

By using the language of vector spaces, you'll be able to describe certain key properties of matrices. The *fundamental subspaces* associated with a matrix A are its *column space* $\mathcal{C}(A)$, its *row space* $\mathcal{R}(A)$, its *null space* $\mathcal{N}(A)$, and its *left null space* $\mathcal{N}(A^\mathsf{T})$. Let's now define these vector spaces and discuss how they help us understand the solutions to the matrix equation $A\vec{x} = \vec{b}$, and the properties of the linear transformation $T_A(\vec{x}) \equiv A\vec{x}$.

Definitions

- V: a *vector space*
- $\vec{v}$: a *vector*. We use the notation $\vec{v} \in V$ to indicate the vector $\vec{v}$ is part of the vector space V.
- W: a *vector subspace*. We use the notation $W \subseteq V$ to indicate the vector space W is a subspace of the vector space V.

- *span*: the span of a set of vectors is the set of vectors that can be constructed as linear combinations of these vectors:

$$\text{span}\{\vec{v}_1, \ldots, \vec{v}_n\} \equiv \{\vec{v} \in V \mid \vec{v} = \alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n,\ \alpha_i \in \mathbb{R}\}.$$

For every matrix $M \in \mathbb{R}^{m\times n}$, we define the following *fundamental vector spaces* associated with the matrix M:

- $\mathcal{R}(M) \subseteq \mathbb{R}^n$: the *row space* of the matrix M consists of all possible linear combinations of the rows of the matrix M.
- $\mathcal{C}(M) \subseteq \mathbb{R}^m$: the *column space* of the matrix M consists of all possible linear combinations of the columns of the matrix M.
- $\mathcal{N}(M) \subseteq \mathbb{R}^n$: the *null space* of M is the set of vectors that go to the zero vector when multiplying M from the right: $\mathcal{N}(M) \equiv \{\vec{v} \in \mathbb{R}^n \mid M\vec{v} = \vec{0}\}$.
- $\mathcal{N}(M^\mathsf{T})$: the *left null space* of M is the set of vectors that go to the zero vector when multiplying M from the left: $\mathcal{N}(M^\mathsf{T}) \equiv \{\vec{w} \in \mathbb{R}^m \mid \vec{w}^\mathsf{T}M = \vec{0}\}$.

Vector space

A vector space V consists of a set of vectors and all possible linear combinations of these vectors. The notion of *all possible linear combinations* is very powerful. In particular, it implies two useful properties. First, vector spaces are *closed under addition*: for all vectors in that space, the sum of two vectors is also a vector in that vector space. Mathematically, we write this as

$$\forall\, \vec{v}_1, \vec{v}_2 \in V, \qquad \vec{v}_1 + \vec{v}_2 \in V.$$

Recall the symbol "$\forall$" is math shorthand for the phrase "for all."

Second, vector spaces are *closed under scalar multiplication*:

$$\forall \alpha \in \mathbb{R} \text{ and } \vec{v} \in V, \qquad \alpha\vec{v} \in V.$$

These two properties codify the essential nature of what a vector space is: a space of vectors that can be added together and scaled by constants.

Span

The *span* operator is a useful shorthand for denoting "the set of all linear combinations" of some set of vectors. This may seem like a weird notion at first, but it will prove very useful for describing vector spaces.

Let's now illustrate how to define vectors spaces using the span operator through some examples. Given a vector $\vec{v}_1 \in V$, define the following vector space:

$$V_1 \equiv \text{span}\{\vec{v}_1\} = \{\vec{v} \in V \mid \vec{v} = \alpha\vec{v}_1 \text{ for some } \alpha \in \mathbb{R}\}.$$

We say V_1 is *spanned* by $\vec{v}_1$, which means any vector in V_1 can be written as a multiple of $\vec{v}_1$. The shape of V_1 is an infinite line.

Given two vectors $\vec{v}_1, \vec{v}_2 \in V$, we define the vector space spanned by these vectors as follows:

$$V_2 \equiv \text{span}\{\vec{v}_1, \vec{v}_2\} = \{\vec{v} \in V \mid \vec{v} = \alpha\vec{v}_1 + \beta\vec{v}_2 \text{ for some } \alpha, \beta \in \mathbb{R}\}.$$

The vector space V_2 contains all vectors that can be written as a linear combination of $\vec{v}_1$ and $\vec{v}_2$. This is a two-dimensional vector space that has the shape of an infinite plane passing through the origin.

Now suppose we're given three vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in V$, such that $\vec{v}_3 = \vec{v}_1 + \vec{v}_2$, and are asked to define the vector space $V_3 \equiv \text{span}\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$. The vector space V_3 is actually the same as V_2; adding the vector $\vec{v}_3$ to the span of $\vec{v}_1$ and $\vec{v}_2$ does not enlarge the vector space because the vector $\vec{v}_3$ is a linear combination of $\vec{v}_1$ and $\vec{v}_2$. Geometrically speaking, the vector $\vec{v}_3$ lies in the same plane as $\vec{v}_1$ and $\vec{v}_2$.

Consider the vector space $V_2' = \text{span}\{\vec{v}_1, \vec{v}_2'\}$, where $\vec{v}_2' = \gamma\vec{v}_1$, for some $\gamma \in \mathbb{R}$. Since $\vec{v}_2'$ is a linear combination of the vector $\vec{v}_1$, the vector space V_2' is one-dimensional. In fact, V_2' is the same as the vector space V_1 defined above: $V_2' = \text{span}\{\vec{v}_1, \vec{v}_2'\} = \text{span}\{\vec{v}_1\} = V_1$.

Vector subspaces

We use the notation $W \subseteq V$ to indicate that W is a *subspace* of V. A *subspace* is a subset of the vectors in the larger space that has a vector space structure. In other words, $W \subseteq V$ if the following conditions are satisfied:

- W is contained in V: for all $\vec{w}$, if $\vec{w} \in W$, then $\vec{w} \in V$.
- W is closed under addition: for all $\vec{w}_1, \vec{w}_2 \in W$, $\vec{w}_1 + \vec{w}_2 \in W$.
- W is closed under scalar multiplication: for all $\vec{w} \in W$, $\alpha\vec{w} \in W$.

Subspaces always contain the zero vector $\vec{0}$. This is implied by the third condition: *any* vector becomes the zero vector when multiplied by the scalar $\alpha = 0$: $\alpha\vec{w} = 0\vec{w} = \vec{0}$.

Subspaces specified by constraints

One way to define a vector subspace W is to start with a larger space V and describe a *constraint* that is satisfied by all vectors in

the subspace W. For example, the xy-plane is defined as the set of vectors $(x, y, z) \in \mathbb{R}^3$ that satisfy the constraint

$$(0, 0, 1) \cdot (x, y, z) = 0.$$

More formally, we define the xy-plane as

$$P_{xy} = \{(x, y, z) \in \mathbb{R}^3 \mid (0, 0, 1) \cdot (x, y, z) = 0\}.$$

Since the vector $\hat{k} \equiv (0, 0, 1)$ is perpendicular to all vectors in the xy-plane, we can describe the xy-plane as "the set of all vectors perpendicular to the vector $\hat{k}$."

Subspaces specified as a span

Another way to represent the xy-plane is to describe it as the span of two linearly independent vectors in the plane:

$$P_{xy} = \text{span}\{\, (1, 0, 0), (0, 1, 0) \,\},$$

which is equivalent to saying:

$$P_{xy} = \{\vec{v} \in \mathbb{R}^3 \mid \vec{v} = \alpha(1, 0, 0) + \beta(0, 1, 0),\ \forall \alpha, \beta \in \mathbb{R}\}.$$

This expression is a *parametrization* of the space P_{xy} with α and β as the two parameters. Each point in the plane is described by a unique pair of parameters (α, β). The parametrization of an m-dimensional vector space requires m parameters.

Subsets vs subspaces

Let's clarify the distinction between the terms *subset* and *subspace*. Assume we're working in a vector space V. A subset of V can be described in the form $S = \{\vec{v} \in V \,|\, \langle \text{conditions} \rangle\}$ and consists of all vectors in V that satisfy the $\langle \text{conditions} \rangle$. A subspace W is a type of subset with a *vector space structure*, meaning it is closed under addition (for all $\vec{w}_1, \vec{w}_2 \in W$, $\vec{w}_1 + \vec{w}_2 \in W$), and closed under scalar multiplication (for all $\vec{w} \in W$, $\alpha\vec{w} \in W$). In linear algebra, the terms *subset* and *subspace* are used somewhat interchangeably, and the same symbol is used to denote both subset ($S \subseteq V$) and subspace ($W \subseteq V$) relationships. When mathematicians refer to some subset as a *subspace*, they're letting you know that you can take arbitrary elements in the set, scale or add them together, and obtain an element of the same set.

To illustrate the difference between a *subset* and a *subspace*, consider the solution sets of a system of equations $A\vec{x} = \vec{b}$ versus the

system of equations $A\vec{x} = \vec{0}$. The solution set of $A\vec{x} = \vec{0}$ is a *vector space* that is called *null space of* A and denoted $\mathcal{N}(A)$. If $\vec{x}_1$ and $\vec{x}_2$ are two solutions to $A\vec{x} = \vec{0}$, then $\alpha\vec{x}_1 + \beta\vec{x}_2$ is also a solution to $A\vec{x} = \vec{0}$. In contrast, the solution set of $A\vec{x} = \vec{b}$ is $\{\vec{c} + \vec{v}_n\}$, for all $\vec{v}_n \in \mathcal{N}(A)$, which is not a subspace of V unless $\vec{c} = \vec{0}$. Observe that if $\vec{x}_1 = \vec{c} + \vec{v}_1$ and $\vec{x}_2 = \vec{c} + \vec{v}_2$ are two solutions to $A\vec{x} = \vec{b}$, their sum is not a solution: $\vec{x}_1 + \vec{x}_2 = 2\vec{c} + \vec{v}_1 + \vec{v}_2 \notin \{\vec{c} + \vec{v}_n\}$.

A real-life situation

You walk into class one day and are caught completely off guard by a surprise quiz—wait, let's make it a mini-exam for emotional effect. Although you've read a chapter or two in the book, you've been "busy" and are totally unprepared for this exam. The first question asks you to "find the solution of the *homogenous* system of equations and the *non-homogenous* system of equations." You rack your brain, but the only association with homogeny that comes to mind is the homogenized milk you had for breakfast. Wait, there's more—the question also asks you to "state whether each of the solutions obtained is a *vector space*." Staring at the page, the words and equations begin to blur as panic sets in.

Don't fear! Look at the problem again. You don't know what the heck a homogenous system of equations is, but you sure as heck know how to solve systems of equations. You solve the given system of equations $A\vec{x} = \vec{b}$ by building the augmented matrix $[\,A\,|\,\vec{b}\,]$ and computing its reduced row echelon form using row operations. You obtain the solution set $\vec{x} = \{\vec{v} \in V \,|\, \vec{v} = \vec{c} + s\vec{v}_n, \forall s \in \mathbb{R}\}$, where $\vec{c}$ is the particular solution and $\vec{v}_n$ is a vector that spans the null space of A.

Next, you ponder the "vector space" part of the question. You notice the solution set to the *system* of equations $A\vec{x} = \vec{b}$ isn't a vector space since it doesn't pass through the origin. However, the solution set to the equation $A\vec{x} = \vec{0}$ *is* a vector space $\{\vec{v} \in V \,|\, \vec{v} = s\vec{v}_n, \forall s \in \mathbb{R}\} = \text{span}\{\vec{v}_n\}$. Suddenly it clicks: a *homogenous* system of equations must be the system of equations $A\vec{x} = \vec{0}$, in which the constants of the right-hand side are all zero. The term *homogenous* kind of makes sense; all the constants of the right-hand side have the same value $b_1 = b_2 = \cdots = 0$. The solution to the non-homogenous system of equations $A\vec{x} = \vec{b}$ is the set $\{\vec{c} + s\vec{v}_n, \forall s \in \mathbb{R}\}$, which is not a vector space. The solution to the homogenous system of equations $A\vec{x} = \vec{0}$ is $\{\vec{v} \in V \,|\, \vec{v} = s\vec{v}_n, \forall s \in \mathbb{R}\}$, which is a vector space. Well done!

Matrix fundamental spaces

We now define four *fundamental vector spaces* associated with a matrix $M \in \mathbb{R}^{m \times n}$.

- The column space $\mathcal{C}(M)$ is the span of the columns of the matrix. The column space consists of all possible output vectors the matrix can produce when multiplied by a vector from the right:

$$\mathcal{C}(M) \equiv \{\vec{w} \in \mathbb{R}^m \mid \vec{w} = M\vec{v} \text{ for some } \vec{v} \in \mathbb{R}^n\}.$$

- The null space $\mathcal{N}(M)$ of a matrix $M \in \mathbb{R}^{m \times n}$ consists of all vectors the matrix M sends to the zero vector:

$$\mathcal{N}(M) \equiv \{\vec{v} \in \mathbb{R}^n \mid M\vec{v} = \vec{0}\}.$$

  The null space is sometimes called the *kernel* of the matrix.
- The row space $\mathcal{R}(M)$ is the span of the rows of the matrix. We obtain linear combinations of the rows by multiplying the matrix *on the left* with an m-dimensional vector:

$$\mathcal{R}(M) \equiv \{\vec{v} \in \mathbb{R}^n \mid \vec{v} = \vec{w}^{\mathsf{T}}M \text{ for some } \vec{w} \in \mathbb{R}^m\}.$$

  Note, we used the transpose $^{\mathsf{T}}$ to transform $\vec{w}$ to a row vector.
- The left null space $\mathcal{N}(M^{\mathsf{T}})$ of a matrix $M \in \mathbb{R}^{m \times n}$ consists of all vectors the matrix M sends to the zero vector when multiplied from the left:

$$\mathcal{N}(M) \equiv \{\vec{w} \in \mathbb{R}^m \mid \vec{w}^{\mathsf{T}}M = \vec{0}\}.$$

These vector spaces are called *fundamental* because they describe important properties of the matrix M. Recall that matrix equations can be used to represent systems of linear equations, and in connection with linear transformations. A solid understanding of fundamental spaces of a matrix leads to a solid understanding of linear equations and linear transformations.

Matrices and systems of linear equations

The null space $\mathcal{N}(M)$ corresponds to the solution of the matrix equation $M\vec{x} = \vec{0}$. If a matrix has a nonempty null space, the system of equations corresponding to $M\vec{x} = \vec{b}$ has an infinite solution set. Indeed, we can write the solution of $M\vec{x} = \vec{b}$ as a *particular solution* $\vec{c}$ plus all possible vectors in the null space of M:

$$\vec{x} = \vec{c} + \text{span}\{\vec{v}_1, \ldots, \vec{v}_k\}, \quad \text{where} \quad \text{span}\{\vec{v}_1, \ldots, \vec{v}_k\} \equiv \mathcal{N}(M).$$

We can verify this claim as follows. Suppose $\vec{x} = \vec{c}$ is a solution to the equation $M\vec{x} = \vec{b}$. Consider the vector $\vec{u} \equiv \vec{c} + \alpha\vec{v}_1 + \cdots + \gamma\vec{v}_k$, which contains $\vec{c}$ and some arbitrary linear combinations of vectors from the null space of M. Observe that $\vec{u}$ is also a solution to the equation $M\vec{x} = \vec{b}$:

$$M\vec{u} = M(\vec{c} + \alpha\vec{v}_1 + \cdots + \gamma\vec{v}_k) = M\vec{c} + \alpha\cancelto{0}{M\vec{v}_1} + \cdots + \gamma\cancelto{0}{M\vec{v}_k} = M\vec{c} = \vec{b}.$$

If the null space of M contains only the zero vector $\{\vec{0}\}$, then the system of equations $M\vec{x} = \vec{b}$ has a unique solution.

Matrices and linear transformations

Matrices can be used to *represent* linear transformations. We postpone the detailed discussion about linear transformations and their representation as matrices until Chapter 6, but we'll discuss the subject here briefly—mainly to introduce an important connection between the column space and the row space of a matrix, and to explain why each matrix has *two* null spaces (what's up with that?).

Matrix-vector and vector-matrix products

A matrix $M \in \mathbb{R}^{m \times n}$ corresponds to not one but *two* linear transformations. Up until now we've focused on the matrix-vector product $M\vec{x} = \vec{y}$, which implements a linear transformation of the form $T_M : \mathbb{R}^n \to \mathbb{R}^m$. In addition to the linear transformation implemented by multiplication from the right, there is also the option of multiplying M by a vector from the left: $\vec{a}^\mathsf{T}M = \vec{b}^\mathsf{T}$, where $\vec{a}^\mathsf{T}$ (the input) is an m-dimensional row vector, and $\vec{b}^\mathsf{T}$ (the output) is an n-dimensional row vector.[1]

The vector-matrix product $\vec{a}^\mathsf{T}M = \vec{b}^\mathsf{T}$ implements the linear transformation of the form $T_{M^\mathsf{T}} : \mathbb{R}^m \to \mathbb{R}^n$. We identify the output of this linear transformation, $\vec{b} = T_{M^\mathsf{T}}(\vec{a})$, with the result of the vector-matrix product $\vec{b}^\mathsf{T} \equiv \vec{a}^\mathsf{T}M$. The linear transformation $T_{M^\mathsf{T}} : \mathbb{R}^m \to \mathbb{R}^n$ is called the *adjoint* of the linear transformation $T_M : \mathbb{R}^n \to \mathbb{R}^m$. Adjoint linear transformations are represented by the same matrix M: T_M is defined as the multiplication of M from the right, while T_{M^T} is defined as multiplication of M from the left.

Let's clarify why we used the notation T_{M^T} to denote the adjoint operation of T_M. We previously used the notation T_A to describe the linear transformation obtained by right multiplication by A: $T_A(\vec{x}) \equiv A\vec{x}$. Instead of creating a new notation for left multiplication, we can

[1] Our convention is to assume vectors are column vectors by default. The transpose operation is required to obtain row vectors.

transform left multiplication into right multiplication by using the transpose operation:

$$\vec{a}^{\mathsf{T}}A = \vec{b}^{\mathsf{T}} \qquad \Rightarrow \qquad \left(\vec{a}^{\mathsf{T}}A\right)^{\mathsf{T}} = \left(\vec{b}^{\mathsf{T}}\right)^{\mathsf{T}} \qquad \Rightarrow \qquad A^{\mathsf{T}}\vec{a} = \vec{b}.$$

We can think of left multiplication by A as right multiplication by A^{T}. This correspondence also explains why we use the notation $\mathcal{N}(M^{\mathsf{T}})$ for the left null space of M; we can rewrite the condition $\vec{w}^{\mathsf{T}}M = \vec{0}^{\mathsf{T}}$ as $M^{\mathsf{T}}\vec{w} = \vec{0}$, so the left null space of M is equivalent to the right null space of M^{T}.

Left and right input spaces

Let's call *left space of M* the set of vectors suitable for multiplying M from the left. Similarly we'll call *right space of M* the set of vectors suitable for multiplying M from the right. If $M \in \mathbb{R}^{m\times n}$, the left space is $\mathbb{R}^m$ and the right space is $\mathbb{R}^n$.

By combining all the vectors in the row space of M and all the vectors in its null space, we obtain the full right space:

$$\mathcal{R}(M) \oplus \mathcal{N}(M) \;=\; \mathbb{R}^n.$$

This means any vector $\vec{v} \in \mathbb{R}^n$ can be written as a sum $\vec{v} = \vec{v}_r + \vec{v}_n$, such that $\vec{v}_r \in \mathcal{R}(M)$ and $\vec{v}_n \in \mathcal{N}(M)$. The symbol $\oplus$ stands for *orthogonal sum*, which means we can pick $\vec{v}_r$ and $\vec{v}_n$ to be orthogonal vectors, $\vec{v}_r \cdot \vec{v}_n = 0$.

If we consider the dimensions involved the above equation, we obtain the following important relation between the dimension of the row space and the null space of a matrix:

$$\dim(\mathcal{R}(M)) + \dim(\mathcal{N}(M)) = n = \dim(\mathbb{R}^n).$$

The n-dimensional right space splits into row-space dimensions and null-space dimensions.

Similarly to the split in the right space, the left-space $\mathbb{R}^m$ decomposes into an orthogonal sum of the column space and the left null space of the matrix:

$$\mathcal{C}(M) \oplus \mathcal{N}(M^{\mathsf{T}}) \;=\; \mathbb{R}^m.$$

If we count the dimensions in this equation, we obtain a relation between the dimension of the column space and the left null space of the matrix: $\dim(\mathcal{C}(M)) + \dim(\mathcal{N}(M^{\mathsf{T}})) = m$.

Matrix rank

The column space and the row space of a matrix have the same dimension. We call this dimension the *rank* of the matrix:

$$\text{rank}(M) \equiv \dim\left(\mathcal{R}(M)\right) = \dim\left(\mathcal{C}(M)\right).$$

The *rank* of M is the number of linearly independent rows in M, which is equal to the number of linearly independent columns in M. The dimension of the null space of M is called the *nullity* of M: $\text{nullity}(M) \equiv \dim(\mathcal{N}(M))$.

Applying this new terminology, we can update our earlier observation about the dimensions of the right fundamental spaces of a matrix:

$$\text{rank}(M) + \text{nullity}(M) = n = \dim(\mathbb{R}^n).$$

This formula is called the *rank–nullity theorem*, and can be used to deduce the rank of a matrix given its nullity, or vice versa.

Summary

Together, $\mathcal{R}(M)$, $\mathcal{N}(M)$, $\mathcal{C}(M)$, and $\mathcal{N}(M^\mathsf{T})$ describe all aspects of the matrix M when multiplied by vectors from the left or the right. Everything we've learned so far about how the matrix M maps vectors between its left and right spaces can be summarized by the following observations:

$$\begin{aligned} \mathcal{C}(M) &\xleftrightarrow{M} \mathcal{R}(M), \\ \vec{0} &\xleftarrow{M} \mathcal{N}(M), \\ \mathcal{N}(M^\mathsf{T}) &\xrightarrow{M} \vec{0}. \end{aligned}$$

Note the zero vector in the second row is $\vec{0} \in \mathbb{R}^m$, while the zero vector in the third row is $\vec{0} \in \mathbb{R}^n$. In Section 6.1, we'll learn how to interpret the fundamental vectors spaces of the matrix M as the input and output spaces of the linear transformations $T_M : \mathbb{R}^n \to \mathbb{R}^m$ and $T_{M^\mathsf{T}} : \mathbb{R}^m \to \mathbb{R}^n$.

Throughout this section, we've referred to the notion of *dimension* of a vector space. The dimension of a vector space is the number of elements in a basis for that vector space. Before we can give a formal definition of the dimension of a vector space, we must review and solidify our understanding of the concept of a basis. We begin with a formal definition of what it means for a set of vectors to be *linearly independent*.

Linear independence

One of the most important ideas in linear algebra is the notion of linear independence. We're often interested in finding vectors that *cannot* be written as linear combinations of others.

The set of vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ is *linearly independent* if the only solution to the equation

$$\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \cdots + \alpha_n\vec{v}_n \;=\; \vec{0}$$

is $\alpha_i = 0$ for all i.

If $\vec{\alpha} = \vec{0}$ is the only solution to the system of equations then none of the vectors $\vec{v}_i$ can be written as a linear combination of the other vectors.

To understand the importance of the "all zero alphas" requirement, let's consider an example where a nonzero solution $\vec{\alpha}$ exists. Suppose the set of vectors $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ satisfy $\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \alpha_3\vec{v}_3 = 0$, with $\alpha_1 = -1$, $\alpha_2 = 1$, and $\alpha_3 = 2$. Then we can write $\vec{v}_1 = 1\vec{v}_2 + 2\vec{v}_3$, which shows that $\vec{v}_1$ is a linear combination of $\vec{v}_2$ and $\vec{v}_3$, hence the vectors are not linearly independent. The strange wording of the definition in terms of an "all zero alphas" solution is required to make the definition of linear independence symmetric. An all zero alphas solution implies that *no* vector can be written as a linear combination of the other vectors.

Basis

To carry out calculations with vectors in a vector space V, we need to know a basis $B = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ for that space. Intuitively, a basis for a vector space is any set of vectors that can serve as a coordinate system for that vector space. A *basis* for an n-dimensional vector space V is a set of n linearly independent vectors in V.

The *dimension* of a vector space is defined as the number of vectors in a basis for that vector space. A basis for an n-dimensional vector space contains exactly n vectors. Any set of less than n vectors would not satisfy the spanning property; any set of more than n vectors from V cannot be linearly independent (see page 196). To form a basis for a vector space, a set of vectors must be "just right": it must contain a sufficient number of vectors (but not too many) so that the coefficients of each vector will be uniquely determined.

The rank–nullity theorem

The relation between the *rank* and the *nullity* of a matrix are so important that it's worth formally stating the **rank–nullity theorem** and showing its proof.

Rank–nullity theorem: *For any matrix* $M \in \mathbb{R}^{m\times n}$*, the following statement holds:*

$$\text{rank}(M) + \text{nullity}(M) = n,$$

where $\text{rank}(M) \equiv \dim\left(\mathcal{R}(M)\right)$ *and* $\text{nullity}(M) \equiv \dim(\mathcal{N}(M))$.

It's not *absolutely* essential that you understand the proof of this theorem, but it's a good idea to read it, as the proof will give you some "exposure" to formal math language used in proofs.

Proof. We must show that the equation $\text{rank}(M) + \text{nullity}(M) = n$ holds. Suppose $\text{rank}(M) = k$ and $B_r = \{\vec{v}_{r1}, \ldots, \vec{v}_{rk}\}$ is a basis for $\mathcal{R}(M)$. Assume furthermore that $\text{nullity}(M) = \ell$ and $B_n = \{\vec{v}_{n1}, \ldots, \vec{v}_{n\ell}\}$ is a basis for the null space of M.

To prove the equation $k + \ell = n$, which relates the dimensions of the row space and the null space to the dimension of $\mathbb{R}^n$, we must prove the equivalent statement about the number of vectors in the bases B_r and B_n. To prove the statement $k + \ell = n$, we must show that the combination of the vectors of B_r and B_n form a basis for $\mathbb{R}^n$.

First, suppose that $k+\ell < n$, and the combined basis vectors from B_r and B_n are insufficient to form a basis for $\mathbb{R}^n$. This means there must exist a vector $\vec{v} \in \mathbb{R}^n$ which *cannot* be expressed in the form:

$$\vec{v} \;=\; v_{r1}\vec{v}_{r1} + \cdots + v_{rk}\vec{v}_{rk} \;+\; v_{n1}\vec{v}_{n1} + \cdots v_{n\ell}\vec{v}_{n\ell}.$$

Since every vector in $\mathcal{R}(M)$ can be expressed in the basis B_r and every vector in $\mathcal{N}(M)$ can be expressed in the basis B_n, it must be that $\vec{v} \notin \mathcal{R}(M)$ and $\vec{v} \notin \mathcal{N}(M)$. However, $\vec{v} \notin \mathcal{R}(M)$ implies $M\vec{v} = \vec{0}$ since each $\vec{v}$ must be orthogonal to each of the rows of M. The fact $M\vec{v} = \vec{0}$ contradicts the fact $\vec{v} \notin \mathcal{N}(M)$, therefore such a vector $\vec{v}$ must not exist; every vector $\vec{v} \in \mathbb{R}^n$ can be written as a linear combination of the combination of the vectors in the bases B_r and B_n.

Now let's analyze the other option $k + \ell > n$. The vectors in the bases B_r and B_n are linearly independent among themselves. All the vectors in B_r are orthogonal to all the vectors in B_n. Since orthogonal implies linearly independent, the combined set of vectors $B_r \cup B_n$ is a linearly independent set. Therefore the equation $k + \ell > n$ is impossible because it would imply the existence of more than n linearly independent vectors in an n-dimensional vector space. $\square$

The **rank–nullity theorem** is important because it "splits" the vectors in the right space of the matrix into two categories: those that lie in its row space and those that lie in its null space. We can use the **rank–nullity theorem** to infer the dimension of the row space of a matrix given the dimension of the null space and vice versa.

Example 1 Suppose the matrix $M \in \mathbb{R}^{m\times n}$ has a trivial null space $\mathcal{N}(M) = \{\vec{0}\}$; then $\text{nullity}(M) \equiv \dim(\mathcal{N}(M)) = 0$ and we can conclude $\text{rank}(M) \equiv \dim(\mathcal{R}(M)) = n$.

Example 2 Consider a matrix $A \in \mathbb{R}^{3\times 6}$. After performing some calculations, which we'll discuss in the next section, we find that one of the rows of the matrix is a linear combination of the other two. Therefore, the row space of A is two-dimensional and $\text{rank}(A) = 2$. From this, we can infer $\text{nullity}(A) = 6-2 = 4$, meaning the null space of A is four-dimensional.

Distilling bases

A basis for an n-dimensional vector space V consist of *exactly* n vectors. Any set of vectors $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ can serve as a basis as long as the vectors are linearly independent and there is exactly n of them.

Sometimes an n-dimensional vector space V will be specified as the span of more than n vectors:

$$V = \text{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m\}, \quad m > n.$$

Since there are $m > n$ of the $\vec{v}$-vectors, they are *too many* to form a basis. We say this set of vectors is *over-complete*. They cannot all be linearly independent since there can be at most n linearly independent vectors in an n-dimensional vector space.

If we want to find a basis for the space V, we'll have to reject some of the vectors. Given the set of vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m\}$, our task is to *distill* a set of n linearly independent vectors $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ from them. We'll learn how to do this in the next section.

Exercises

5.5 Vector space techniques

In this section, we'll learn how to "distill" a basis for any vector space; an important procedure for characterizing vector spaces. Actually, the procedure is not new—it's really an application of the Gauss–Jordan elimination procedure we saw in Section 4.1.

Starting from a set of vectors that are not linearly independent, we can write them as the rows of a matrix, and then perform *row operations* on this matrix until we find the reduced row echelon form of the matrix. Since row operations do not change the row space of a matrix, the nonzero rows in the final RREF of the matrix span the same space as the original set of vectors. The rows in the RREF of the matrix will be linearly independent so they form a basis.

The ability to distill a basis is important when characterizing any vector space. The basis serves as the coordinate system for that vector space, and the number of vectors in a basis tells us the dimension of the vector space. For this reason, we'll spend an entire section learning how to distill bases for various vector spaces.

Finding a basis

Suppose the vector subspace V is defined as the span of m vectors $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\}$, which are not necessarily linearly independent:

$$V \equiv \text{span}\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\}.$$

Our task is to find a basis for V. We're looking for an alternate description of the vector space V as

$$V = \text{span}\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\},$$

such that the vectors $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ will be linearly independent.

One way to accomplish this task is to write the vectors $\vec{u}_i$ as the rows of a matrix M. By this construction, the space V corresponds to the *row space* of the matrix M, denoted $\mathcal{R}(M)$. We can then use standard *row operations* to bring the matrix into its reduced row echelon form. Applying row operations to a matrix does not change its row space: $\mathcal{R}(M) = \mathcal{R}(\text{rref}(M))$. By transforming the matrix into its RREF, we're able to see which of the rows are linearly independent and can thus serve as basis vectors:

$$\begin{bmatrix} — & \vec{u}_1 & — \\ — & \vec{u}_2 & — \\ — & \vec{u}_3 & — \\ & \vdots & \\ — & \vec{u}_m & — \end{bmatrix} \quad - \text{ G-J elimination} \rightarrow \quad \begin{bmatrix} — & \vec{e}_1 & — \\ & \vdots & \\ — & \vec{e}_n & — \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The nonzero rows in the RREF of the matrix form a set of linearly independent vectors $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ that span the vector space V. The linearly dependent vectors have been reduced to rows of zeros.

The above process is called "finding a basis" or "distilling a basis" and it's important you understand how to carry out this procedure.

Even more important is that you understand *why* we'd want to distill a basis in the first place! By the end of the Gauss–Jordan procedure, we obtain a description of the same vector space V in terms of a new set of vectors. Why is it better to describe the vector space V in terms of the vectors $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$, rather than in terms of $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\}$?

I'll tell you exactly why. We prefer to use the basis $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ to characterize the vector space V because there exists a one-to-one correspondence between each vector $\vec{v} \in V$ and the coefficients $v_1, v_2, \ldots, v_n$ in the linear combination

$$\vec{v} \;=\; v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n.$$

Using the basis $B \equiv \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ allows us to represent each vector $\vec{v} \in V$ as a unique list of coefficients $(v_1, v_2, \ldots, v_n)_B$.

This would not be possible if we used the vectors $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\}$ to describe the vector space V. Since the vectors $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\}$ are not linearly independent, the same vector $\vec{v}$ could be represented by many different linear combination of the form

$$\vec{v} \;=\; v_1'\vec{u}_1 + v_2'\vec{u}_2 + \cdots + v_m'\vec{u}_m.$$

We cannot identify $\vec{v}$ with a *unique* set of coefficients $v_1', v_2', \ldots, v_m'$, therefore vectors are not represented faithfully by their coefficients.

Another reason we prefer to describe V in terms of a basis is because we can immediately see the vector space V is n-dimensional, since there are n vectors in the basis for V.

Definitions

- $B = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$. A *basis* for an n-dimensional vector space S is a set of n linearly independent vectors that span S. Any vector $\vec{v} \in S$ can be written as a linear combination of the basis elements:
$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n.$$
A basis for an n-dimensional vector space contains exactly n vectors.
- $\dim(S)$: the dimension of the vector space S is equal to the number of elements in a basis for S.

Recall the four *fundamental spaces* of a matrix $M \in \mathbb{R}^{m\times n}$ we defined in the previous section:

- $\mathcal{R}(M) \subseteq \mathbb{R}^n$: the *row space* of the matrix M that consists of all possible linear combinations of the rows of the matrix M.

- $\mathcal{N}(M) \subseteq \mathbb{R}^n$: the *null space* of the matrix contains all the vectors that become the zero vector when multiplied by M:

$$\mathcal{N}(M) \equiv \{ \vec{v} \in \mathbb{R}^n \ | \ M\vec{v} = \vec{0} \}.$$

- $\mathcal{C}(M) \subseteq \mathbb{R}^m$: the *column space* of the matrix M that consists of all possible linear combinations of the columns of the matrix M.
- $\mathcal{N}(M^\mathsf{T}) \subseteq \mathbb{R}^m$: the *left null space* of the matrix contains all the vectors that become the zero vector when multiplying M from the left:

$$\mathcal{N}(M^\mathsf{T}) \equiv \{\vec{w} \in \mathbb{R}^m \mid \vec{w}^\mathsf{T} M = \vec{0}^\mathsf{T}\}.$$

Bases for the fundamental spaces of matrices

Performing the Gauss–Jordan elimination procedure on a matrix A has the effect of "distilling" a basis for its row space $\mathcal{R}(A)$. How do we find bases for the other fundamental spaces of a matrix? In this subsection we'll learn about a useful shortcut for computing bases for the column space $\mathcal{C}(A)$ and the null space $\mathcal{N}(A)$ of a matrix, starting from the reduced row echelon form of the matrix. Sorry, there is no shortcut for finding the left null space—we'll have to use the transpose operation to obtain A^T and then find its null space $\mathcal{N}(A^\mathsf{T})$.

Pay careful attention to the locations of the pivots (leading ones) in the RREF of A because they play an important role in the procedures described below.

Basis for the row space

The row space $\mathcal{R}(A)$ of a matrix A is defined as the space of all vectors that can be written as linear combinations of the rows of A. To find a basis for $\mathcal{R}(A)$, we use the Gauss–Jordan elimination procedure:

1. Perform row operations to find the RREF of A.
2. The nonzero rows in the RREF of A form a basis for $\mathcal{R}(A)$.

Basis for the column space

To find a basis for the column space $\mathcal{C}(A)$ of a matrix A, we need to determine which columns of A are linearly independent. To find the linearly independent columns of A, follow these steps:

1. Perform row operations to find the RREF of A.
2. Identify the columns which contain the pivots (leading ones).
3. The corresponding columns **in the original matrix** A form a basis for the column space of A.

This procedure works because elementary row operations do not change the independence relations between the columns of the matrix. If two columns are linearly independent in the RREF of A, then these columns were also linearly independent in the original matrix A.

Note that the column space of the matrix A corresponds to the row space of the matrix transposed A^T. Thus, another procedure for finding a basis for the column space of a matrix A is to use the find-a-basis-for-the-row-space procedure on A^T.

Basis for the null space

The null space $\mathcal{N}(A)$ of a matrix $A \in \mathbb{R}^{m\times n}$ is

$$\mathcal{N}(A) = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{0}\}.$$

The vectors in the null space are orthogonal to the row space of the matrix A.

The null space of A is the *solution* of the equation $A\vec{x} = \vec{0}$. You should already be familiar with the procedure for finding the solution of systems of equations from Section 4.1. The steps of the procedure are:

1. Perform row operations to find the RREF of A.
2. Identify the columns that do not contain a leading one. These columns correspond to *free variables* of the solution. For example, consider a matrix whose reduced row echelon form is

$$\text{rref}(A) = \begin{bmatrix} \mathbf{1} & 2 & 0 & 0 \\ 0 & 0 & \mathbf{1} & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

   The second column and the fourth column do not contain leading ones (pivots), so they correspond to free variables, which are customarily called s, t, r, etc. We're looking for a vector with two free variables: $(x_1, s, x_3, t)^\mathsf{T}$.
3. Rewrite the null space problem as a system of equations:

$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ s \\ x_3 \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} 1x_1 + 2s &= 0 \\ 1x_3 - 3t &= 0 \\ 0 &= 0\,. \end{aligned}$$

   We can express the unknowns x_1 and x_3 in terms of the free variables s and t: $x_1 = -2s$ and $x_3 = 3t$. The vectors in the null space are of the form $(-2s, s, 3t, t)^\mathsf{T}$, for all $s, t \in \mathbb{R}$. We

can rewrite this expression by splitting it into an s-part and a t-part:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2s \\ s \\ 3t \\ t \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix} s + \begin{bmatrix} 0 \\ 0 \\ 3 \\ 1 \end{bmatrix} t.$$

4. The direction vectors associated with each free variable form a basis for the null space of the matrix A:

$$\mathcal{N}(A) = \left\{ \begin{bmatrix} -2s \\ s \\ 3t \\ t \end{bmatrix}, \forall s, t \in \mathbb{R} \right\} = \text{span} \left\{ \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 3 \\ 1 \end{bmatrix} \right\}.$$

Verify that the matrix A multiplied by any vector from its null space produces a zero vector.

Examples

Let's check out a couple examples that illustrate the procedures for finding bases for $\mathcal{R}(A)$, $\mathcal{C}(A)$, and $\mathcal{N}(A)$. It's important that you become proficient at these "find a basis" tasks because they often appear on homework assignments and exams.

Example 1 Find a basis for the row space, the column space, and the null space of the matrix:

$$A = \begin{bmatrix} 4 & -4 & 0 \\ 1 & 1 & -2 \\ 2 & -6 & 4 \end{bmatrix}.$$

The first steps toward finding the row space, column space, and the null space of a matrix all require calculating the RREF of the matrix, so this is what we'll do first.

- Let's focus on the first column. To create a pivot in the top left corner, we divide the first row by 4, denoted $R_1 \leftarrow \frac{1}{4} R_1$:

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & -2 \\ 2 & -6 & 4 \end{bmatrix}.$$

- We use this pivot to clear the numbers on the second and third rows by performing $R_2 \leftarrow R_2 - R_1$ and $R_3 \leftarrow R_3 - 2R_1$:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 2 & -2 \\ 0 & -4 & 4 \end{bmatrix}.$$

- We can create a pivot in the second row if we divide it by 2, denoted $R_2 \leftarrow \frac{1}{2}R_2$:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -4 & 4 \end{bmatrix}.$$

- We now clear the entry below the pivot using $R_3 \leftarrow R_3 + 4R_2$:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}.$$

- The final simplification step is to clear the -1 in the first row using $R_1 \leftarrow R_1 + R_2$:

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Now that we have the RREF of the matrix, we can answer the questions like professionals.

Before we find bases for the fundamental spaces of A, let's first do some basic dimension counting. Observe that the matrix has just two pivots. We say $\text{rank}(A) = 2$. This means both the row space and the column spaces are two-dimensional.

Recall the equation $n = \text{rank}(A) + \text{nullity}(A)$, which we saw in the previous section. The right space $\mathbb{R}^3$ splits into two types of vectors: those in the row space of A and those in the null space. Since we know the row space is two-dimensional, we can deduce the dimension of the null space: $\text{nullity}(A) \equiv \dim(\mathcal{N}(A)) = n - \text{rank}(A) = 3 - 2 = 1$.

Now let's answer the questions posed in the problem.

- The row space of A consists of the two nonzero vectors in the RREF of A:

$$\mathcal{R}(A) = \text{span}\{(1, 0, -1), (0, 1, -1)\}.$$

- To find the column space of A, observe that the first and the second columns contain the pivots in the RREF of A. If they do, then the first two columns of the original matrix A form a basis for the column space of A:

$$\mathcal{C}(A) = \text{span}\left\{ \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -4 \\ 1 \\ -6 \end{bmatrix} \right\}.$$

- Let's now find an expression for the null space of A. First, observe that the third column does not contain a pivot. No pivot indicates that the third column corresponds to a free variable; it can take on any value $x_3 = t, \quad t \in \mathbb{R}$. We want to give a description of all vectors $(x_1, x_2, t)^\mathsf{T}$ that satisfy the system of equations:

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} 1x_1 - 1t &= 0 \\ 1x_2 - 1t &= 0 \\ 0 &= 0\,. \end{aligned}$$

We find $x_1 = t$ and $x_2 = t$ and obtain the following final expression for the null space:

$$\mathcal{N}(A) = \left\{ \begin{bmatrix} t \\ t \\ t \end{bmatrix}, \quad t \in \mathbb{R} \right\} = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

The null space of A is one-dimensional and consists of all multiples of the vector $(1, 1, 1)^\mathsf{T}$.

Example 2 Find a basis for the row space, column space, and null space of the matrix:

$$B = \begin{bmatrix} 1 & 3 & 1 & 4 \\ 2 & 7 & 3 & 9 \\ 1 & 5 & 3 & 1 \\ 1 & 2 & 0 & 8 \end{bmatrix}.$$

First we find the reduced row echelon form of the matrix B:

$$\sim \begin{bmatrix} 1 & 3 & 1 & 4 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & -3 \\ 0 & -1 & -1 & 4 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & -2 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & -5 \\ 0 & 0 & 0 & 5 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & -2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

As in the previous example, we begin by calculating the dimensions of the subspaces. The rank of this matrix is three, so the column space and the row space will be three-dimensional. Since the right space is $\mathbb{R}^4$, this leaves one dimension for the null space. Next, let's find the fundamental spaces for the matrix B.

- The row space of B consists of the three nonzero vectors in the RREF of B:

$$\mathcal{R}(B) = \text{span}\{\ (1, 0, -2, 0), (0, 1, 1, 0), (0, 0, 0, 1)\}.$$

- The column space of B is spanned by the first, second and fourth columns of B since these columns contain the leading ones in the RREF of B:

$$\mathcal{C}(B) = \text{span}\left\{ \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 7 \\ 5 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 9 \\ 1 \\ 8 \end{bmatrix} \right\}.$$

- The third column lacks a leading one, so it corresponds to a free variable $x_3 = t$, $t \in \mathbb{R}$. The null space of B is the set of vectors $(x_1, x_2, t, x_4)^\mathsf{T}$ such that:

$$\begin{bmatrix} 1 & 0 & -2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ t \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} 1x_1 - 2t &= 0 \\ 1x_2 + 1t &= 0 \\ x_4 &= 0 \\ 0 &= 0\,. \end{aligned}$$

We find the values of x_1, x_2, and x_4 in terms of t and obtain

$$\mathcal{N}(B) = \left\{ \begin{bmatrix} 2t \\ -t \\ t \\ 0 \end{bmatrix}, \ t \in \mathbb{R} \right\} = \text{span}\left\{ \begin{bmatrix} 2 \\ -1 \\ 1 \\ 0 \end{bmatrix} \right\}.$$

Discussion

Dimensions

For an $m \times n$ matrix $M \in \mathbb{R}^{m \times n}$ the row space and the column space consist of vectors with n components, while the column space and the left null space consist of vectors with m components.

Don't confuse the number of components of vectors in a vector space with the *dimension* of the space. Suppose we're given a matrix with five rows and ten columns $M \in \mathbb{R}^{5 \times 10}$ and the RREF of M contains 3 pivots. We say the *rank* of the matrix is 3, which means the row space of M is 3-dimensional. A basis for the row space of M contains 3 vectors, each vector having 10 components. The null space of the matrix is 7-dimensional ($10 - 3 = 7$) and consist of vectors with 10 components. The column space of the matrix is also be 3-dimensional ($\mathcal{R}(M) = \mathcal{C}(M)$). A basis for the column space of M consists of 3 vectors with 5 components. The left null space of M is 2-dimensional ($5 - 3 = 2$) and is spanned by vectors with 5 components.

Importance of bases

The procedures for identifying bases are somewhat technical and potentially boring, but they are of great practical importance. To illustrate the importance of a basis, consider a scenario in which you're given a description of the xy-plane P_{xy} as the span of *three* vectors:

$$P_{xy} = \text{span}\{(1,0,0),(0,1,0),(1,1,0)\}.$$

The above definition of P_{xy} says that any point $p \in P_{xy}$ can be written as a linear combination of the form

$$p = a(1,0,0) + b(0,1,0) + c(1,1,0),$$

for some coefficients (a,b,c). This representation of P_{xy} is misleading. It might make us think (erroneously) that P_{xy} is three-dimensional, since we need three coefficients (a,b,c) to describe points in P_{xy}.

Do we really need *three* coefficients to describe any $p \in P_{xy}$? No, we don't. Two vectors are sufficient: $(1,0,0)$ and $(0,1,0)$, for example. The same point p described above can be written as:

$$p = \underbrace{(a+c)}_{\alpha}(1,0,0) + \underbrace{(b+c)}_{\beta}(0,1,0) = \alpha(1,0,0) + \beta(0,1,0).$$

Note the point is described in terms of *two* coefficients (α,β). The vector $(1,1,0)$ is not *necessary* for the description of points in P_{xy}. The vector $(1,1,0)$ is redundant because it can be expressed in terms of the vectors $(1,0,0)$ and $(0,1,0)$. By getting rid of the redundant vector, we obtain a description of P_{xy} in terms of a basis:

$$P_{xy} = \text{span}\{(1,0,0),(0,1,0)\}.$$

Recall that the requirement for a basis B for a space V is that it be made of linearly independent vectors and that it span the space V. The vectors $\{(1,0,0),(0,1,0)\}$ are sufficient to represent any vector in P_{xy}, *and* these vectors are linearly independent. We can correctly conclude that the space P_{xy} is two-dimensional. If someone asks you "how do you know that P_{xy} is two-dimensional?," say "because its basis contains two vectors."

Exercises

E5.6 Consider the following matrix:

$$A = \begin{bmatrix} 1 & 3 & 3 & 3 \\ 2 & 6 & 7 & 6 \\ 3 & 9 & 9 & 10 \end{bmatrix}.$$

Find the RREF of A and bases for $\mathcal{R}(A)$, $\mathcal{C}(A)$, and $\mathcal{N}(A)$.

E5.7 Find the null space $\mathcal{N}(A)$ of the following matrix A:

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 4 & 0 & 0 & -2 \\ 0 & 2 & -2 & -1 \end{bmatrix}.$$

5.6 Geometrical problems

We saw computations
now geometry prove you can imagine

P5.1 Find intersection of two lines: a) ℓ_1: $2x+y=4$ and ℓ_2: $3x-2y=-1$; b) ℓ_1: $y+x=2$ and ℓ_2: $2x+2y=4$; c) ℓ_1: $y+x=2$ and ℓ_2: $y-x=0$.

P5.2 Find an intersection line between two planes: a) P_1: $3x-2y-z=2$ and P_2: $x+2y+z=0$; d) P_3: $2x+y-z=0$ and P_4: $x+2y+x=3$.

P5.3 Find if the planes are parallel, perpendicular or neither: a) P_1: $x-y-z=0$ and P_2: $2z-2y-2z=4$; b) P_3: $3x+2y=1$ and P_4: $y-z=0$; c) P_5: $x-2y+z=5$ and P_6: $x+y+z=3$.

P5.4 Find the distance from the point $q=(2,3,5)$ to the plane P : $2x+y-2z=0$.

P5.5 Find distance between points: a) $p=(4,5,3)$ and $q=(1,1,1)$; b) $m=(4,-2,0)$ and $n=(0,1,0)$; c) $r=(1,0,1)$ and $s=(-1,1,-1)$; d) $p=(2,1,2)$ and $j=(1,-2,-1)$.

P5.6 Find the general equation of the plane that passes through the points $q=(1,3,0)$, $r=(0,2,1)$, and $s=(1,1,1)$.

P5.7 Find the symmetric equation of the line ℓ described by the equation

$$x=2t-3, \quad y=-4t+1, \quad z=-t.$$

P5.8 Given two vectors $\vec{u}=(2,1,-1)$ and $\vec{v}=(1,1,1)$, find the projection of $\vec{v}$ onto $\vec{u}$, and then the projection of $\vec{u}$ onto $\vec{v}$.

P5.9 Find a projection of $\vec{v}=(3,4,1)$ onto the plane $P: 2x-y+4z=4$.

P5.10 Find the component of the vector $\vec{u}=(-2,1,1)$ that is perpendicular to the plane P formed by the points $m=(2,4,1)$, $s=(1,-2,4)$, and $r=(0,4,0)$.

P5.11 Find the distance between the line $\ell: \{x=1+2t, y=-3+t, z=2\}$, and the plane $P: -x+2y+2z=4$.

P5.12 An $m \times n$ matrix A is upper triangular if all entries lying below the main diagonal are zero, that is, if $A_{ij}=0$ whenever $i>j$. Prove that upper triangular matrices form a subspace of $\mathbb{R}^{m\times n}$.

P5.13 Prove that diagonal matrices are symmetric matrices.

P5.14 Let $\vec{u}$ and $\vec{v}$ be distinct vectors from the vector space V. Show that if $\{\vec{u}, \vec{v}\}$ is a basis for V and a and b are nonzero scalars, then both $\{\vec{u} + \vec{v}, a\vec{u}\}$ and $\{a\vec{u}, b\vec{v}\}$ are also bases for V.

P5.15 Suppose that $S = \{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ is linearly independent and $\vec{w}_1 = \vec{v}_1 + \vec{v}_2 + \vec{v}_3, \vec{w}_2 = \vec{v}_2 + \vec{v}_3$ and $\vec{w}_3 = \vec{v}_3$. Show that $T = \{\vec{w}_1, \vec{w}_2, \vec{w}_3\}$ is linearly independent.

P5.16 Compute the product matrix of the following thee matrices:

$$A = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \sqrt{x^2+1} - x & 0 \\ 0 & \sqrt{x^2+1} + x \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where $\phi = -\frac{\pi}{4} - \frac{1}{2}\arctan x$, and $\theta = \frac{\pi}{4} - \frac{1}{2}\arctan x$.

Chapter 6

Linear transformations

Linear transformations are a central idea of linear algebra—they form the cornerstone that connects all the seemingly unrelated concepts we've studied so far. We previously introduced linear transformations, informally describing them as "vector functions." In this chapter, we'll formally define linear transformations, describe their properties, and discuss their applications.

In Section 6.2, we'll learn how matrices can be used to *represent* linear transformations. We'll show the matrix representations of important types of linear transformations like projections, reflections, and rotations. Section 6.3 discusses the relation between bases and matrix representations. We'll learn how the bases chosen for the input and output spaces determine the coefficients of matrix representations. The same linear transformation corresponds to different matrix representations, depending on the choice of bases for the input and output spaces. Section 6.4 discusses and characterizes the class of *invertible linear transformations*. This section will serve to connect several topics we covered previously: linear transformations, matrix representations, and the fundamental vector spaces of matrices.

6.1 Linear transformations

Linear transformations take vectors as inputs and produce vectors as outputs. A transformation T that takes n-dimensional vectors as inputs and produces m-dimensional vectors as outputs is denoted $T\colon \mathbb{R}^n \to \mathbb{R}^m$.

The class of linear transformations includes most of the useful transformations of analytical geometry: stretchings, projections, reflections, rotations, and combinations of these. Since linear transformations describe and model many real-world phenomena in physics,

chemistry, biology, and computer science, it's worthwhile to learn the theory behind them.

Concepts

Linear transformations are mappings between *vector inputs* and *vector outputs*. The following concepts describe the input and output spaces:

- V: the input vector space of T
- W: the output space of T
- $\dim(U)$: the dimension of the vector space U
- $T : V \to W$: a linear transformation that takes vectors $\vec{v} \in V$ as inputs and produces vectors $\vec{w} \in W$ as outputs. The notation $T(\vec{v}) = \vec{w}$ describes T acting on $\vec{v}$ to produce the output $\vec{w}$.

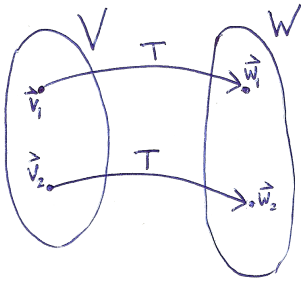


Figure 6.1: An illustration of the linear transformation $T : V \to W$.

- $\text{Im}(T)$: the *image space* of the linear transformation T is the set of vectors that T can output for some input $\vec{v} \in V$. The mathematical definition of the image space is

$$\text{Im}(T) \equiv \{\vec{w} \in W \mid \vec{w} = T(\vec{v}), \text{ for some } \vec{v} \in V\} \quad \subseteq \quad W.$$

  The image space is the vector equivalent of the *image set* of a single-variable function $\text{Im}(f) \equiv \{y \in \mathbb{R} \,|\, y = f(x), \forall x \in \mathbb{R}\}$.

- $\text{Ker}(T)$: the *kernel* of the linear transformation T; the set of vectors mapped to the zero vector by T. The mathematical definition of the kernel is

$$\text{Ker}(T) \equiv \{\vec{v} \in V \mid T(\vec{v}) = \vec{0}\} \quad \subseteq \quad V.$$

  The kernel of a linear transformation is the vector equivalent of the roots of a function: $\{x \in \mathbb{R} \,|\, f(x) = 0\}$.

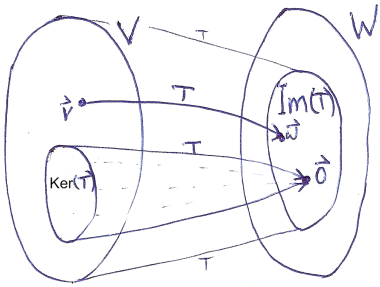


Figure 6.2: Two key properties of a linear transformation $T : V \to W$; its kernel $\text{Ker}(T) \subseteq V$, and its image space $\text{Im}(T) \subseteq W$.

Example The linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^3$ is defined by the equation $T(x, y) = (x, y, x + y)$. Applying T to the input vector $(1, 0)$ produces the output vector $(1, 0, 1 + 0) = (1, 0, 1)$. Applying T to the input vector $(3, 4)$ produces the output vector $(3, 4, 7)$.

The kernel of T contains only the zero vector $\text{Ker}(T) = \{\vec{0}\}$. The image space of T is a two-dimensional subspace of the output space $\mathbb{R}^3$, namely $\text{Im}(T) = \text{span}\{(1, 0, 1), (0, 1, 1)\} \subseteq \mathbb{R}^3$.

Matrix representations

Given bases for the input and output spaces of a linear transformation T, the transformation's action on vectors can be represented as a matrix-vector product:

- $B_V = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$: a basis for the input vector space V
- $B_W = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_m\}$: a basis for the output vector space W
- $M_T \in \mathbb{R}^{m \times n}$: a matrix representation of the linear transformation T:

$$\vec{w} = T(\vec{v}) \qquad \Leftrightarrow \qquad \vec{w} = M_T \vec{v}.$$

  To be precise, we denote the matrix representation as ${}_{B_W}[M_T]_{B_V}$ to show it depends on the input and output bases.
- $\mathcal{C}(M_T)$: the *column space* of a matrix M_T
- $\mathcal{N}(M_T)$: the *null space* a matrix M_T

Properties of linear transformations

We'll start with the feature of linear transformations that makes them suitable for modelling a wide range of phenomena in science, engineering, business, and computing.

Linearity

The fundamental property of linear transformations is—you guessed it—their *linearity*. If $\vec{v}_1$ and $\vec{v}_2$ are two input vectors and α and β are two constants, then

$$T(\alpha\vec{v}_1 + \beta\vec{v}_2) = \alpha T(\vec{v}_1) + \beta T(\vec{v}_2) = \alpha\vec{w}_1 + \beta\vec{w}_2,$$

where $\vec{w}_1 = T(\vec{v}_1)$ and $\vec{w}_2 = T(\vec{v}_2)$.

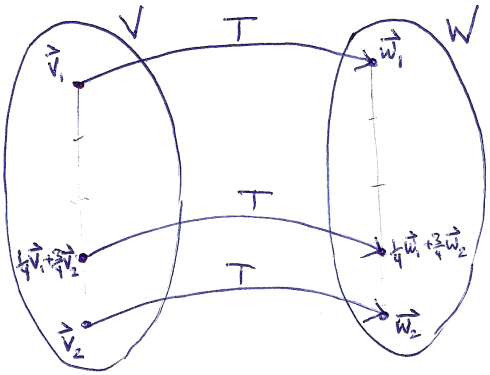


Figure 6.3: A linear transformation T maps the linear combination of inputs $\frac{1}{4}\vec{v}_1 + \frac{3}{4}\vec{v}_2$ to the linear combination of outputs $\frac{1}{4}\vec{w}_1 + \frac{3}{4}\vec{w}_2$.

Linear transformations map a linear combination of inputs to the same linear combination of outputs. If you know the outputs of T for the inputs $\vec{v}_1$ and $\vec{v}_2$, you can deduce the output T for any linear combination of the vectors $\vec{v}_1$ and $\vec{v}_2$ by computing the appropriate linear combination of the outputs $T(\vec{v}_1)$ and $T(\vec{v}_2)$. This is perhaps the most important idea in linear algebra. This is the *linear* that we're referring to when we talk about *linear algebra*. Linear algebra is not about lines, but about mathematical transformations that map linear combination of inputs to the same linear combination of outputs.

In this chapter we'll study various aspects and properties of linear transformations, these abstract objects that map input vectors to output vectors. The fact that linear transformations map linear combinations of inputs to corresponding linear combinations of its outputs will be of central importance in many calculations and proofs. Make a good note and store a mental image of the example shown in Figure 6.3.

Linear transformations as black boxes

Suppose someone gives you a black box that implements the linear transformation T. While you can't look inside the box to see how

T acts, you can *probe* the transformation by choosing various input vectors and observing what comes out.

Assume the linear transformation T is of the form $T\colon \mathbb{R}^n \to \mathbb{R}^m$. By probing this transformation with n vectors of a basis for the input space and observing the outputs, you can characterize the transformation T completely.

To see why this is true, consider a basis $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ for the n-dimensional input space $V = \mathbb{R}^n$. To characterize T, input each of the n basis vectors $\vec{e}_i$ into the black box and record the $T(\vec{e}_i)$ that comes out.

Any input vector $\vec{v}$ can be written as a linear combination of the basis vectors:

$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n.$$

Using these observations and the linearity of T, we can predict the output of T for this vector:

$$T(\vec{v}) = v_1T(\vec{e}_1) + v_2T(\vec{e}_2) + \cdots + v_nT(\vec{e}_n).$$

This black box model is used in many areas of science and is one of the most important ideas in linear algebra. The transformation T could be the description of a chemical process, an electrical circuit, or some phenomenon in biology. As long as we know that T is (or can be approximated by) a linear transformation, we can describe it completely by "probing" it with a small number of inputs. This is in contrast to characterizing non-linear transformations, which correspond to arbitrarily complex input-output relationships and require significantly more probing.

Input and output spaces

Consider the linear transformation T from n-vectors to m-vectors:

$$T\colon \mathbb{R}^n \to \mathbb{R}^m.$$

The *domain* of the function T is $\mathbb{R}^n$ and its *codomain* is $\mathbb{R}^m$.

The *image space* $\text{Im}(T)$ consists of all possible outputs of the transformation T. The image space is a subset of the output space, $\text{Im}(T) \subseteq \mathbb{R}^m$. A linear transformation T whose image space is equal to its codomain ($\text{Im}(T) = \mathbb{R}^m$) is called *surjective* or *onto*. Recall that a function is surjective if it covers the entire output set.

The *kernel* of T is the subspace of the domain $\mathbb{R}^n$ that is mapped to the zero vector by T: $\text{Ker}(T) \equiv \{\vec{v} \in \mathbb{R}^n \mid T(\vec{v}) = \vec{0}\}$. A linear transformation with an empty kernel $\text{Ker}(T) = \{\vec{0}\}$ is called *injective*. Injective transformations map different inputs to different outputs.

If a linear transformation T is both injective and surjective it is called *bijective*. In this case, T is a *one-to-one correspondence* between the input vector space and the output vector space.

Note the terminology used to characterize linear transformations (injective, surjective, and bijective) is the same as the terminology used to characterize functions in Section 1.8. Indeed, linear transformations are simply vector functions, so we can use the same terminology. The concepts of image space and kernel are illustrated in Figure 6.4.

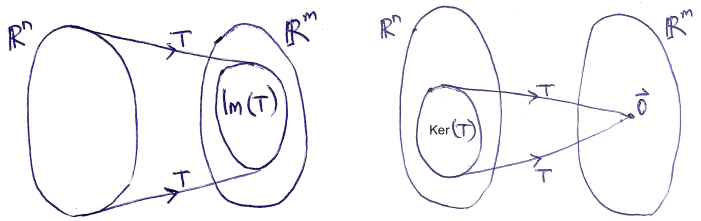


Figure 6.4: Pictorial representations of the image space $\mathrm{Im}(T)$ and the kernel $\mathrm{Ker}(T)$ of a linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$. The image space is the set of all possible outputs of T. The kernel of T is the set of inputs that T maps to the zero vector.

Observation The dimensions of the input space and the output space of a bijective linear transformation must be the same. Indeed, if $T : \mathbb{R}^n \to \mathbb{R}^m$ is bijective then it is both injective and surjective. Since T is surjective, the input space must be at least as large as the output space $n \geq m$. Since T is injective, the output space must be larger or equal to the input space $m \geq n$. Combining these observations, we find that if $T : \mathbb{R}^n \to \mathbb{R}^m$ is bijective then $m = n$.

Example 2 Consider the linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^2$ defined by the equation $T(x, y, z) = (x, z)$. Find the kernel and the image space of T. Is T injective? Is T surjective?

The action of T is to delete the y-components of inputs. Any vector that has only a y-component will be sent to the zero vector. We have $\mathrm{Ker}(T) = \mathrm{span}\{(0, 1, 0)\}$. The image space is $\mathrm{Im}(T) = \mathbb{R}^2$. The transformation T is not injective. As an explicit example proving T is not injective, observe that $T(0, 1, 0) = T(0, 2, 0)$ but $(0, 1, 0) \neq (0, 2, 0)$. Since $\mathrm{Im}(T)$ is equal to the codomain $\mathbb{R}^2$, T is surjective.

Linear transformations as matrix multiplications

An important relationship exists between linear transformations and matrices. If you fix a basis for the input vector space and a basis

for the output vector space, a linear transformation $T(\vec{v}) = \vec{w}$ can be represented as matrix multiplication $M_T\vec{v} = \vec{w}$ for some matrix M_T:

$$\vec{w} = T(\vec{v}) \qquad \Leftrightarrow \qquad \vec{w} = M_T\vec{v}.$$

Using this equivalence, we can re-interpret several properties of matrices as properties of linear transformations. The equivalence is useful in the other direction too, since it allows us to use the language of linear transformations to talk about the properties of matrices.

The idea of representing the action of a linear transformation as a matrix product is extremely important since it transforms the *abstract* description of the transformation T into the *concrete* one: "take the input vector $\vec{v}$ and multiply it from the right by the matrix M_T."

Example 3 We'll now illustrate the "linear transformation $\Leftrightarrow$ matrix-product" equivalence with an example. Define $\Pi_{P_{xy}}$ to be the *orthogonal projection* onto the xy-plane P_{xy}. In words, the action of this projection is to zero-out the z-component of input vectors. The matrix that corresponds to this projection is

$$T\left(\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\right) = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \qquad \Leftrightarrow \qquad M_T\vec{v} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}.$$

Finding the matrix

In order to find the matrix representation of any linear transformation $T\colon \mathbb{R}^n \to \mathbb{R}^m$, it is sufficient to probe T with the n vectors in the standard basis for $\mathbb{R}^n$:

$$\hat{e}_1 \equiv \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \hat{e}_2 \equiv \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad \hat{e}_n \equiv \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

To obtain M_T, we combine the outputs $T(\hat{e}_1)$, $T(\hat{e}_2)$, ..., $T(\hat{e}_n)$ as the *columns* of a matrix:

$$M_T = \begin{bmatrix} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \ldots & T(\hat{e}_n) \\ | & | & & | \end{bmatrix}.$$

Observe that the matrix constructed in this way has the right dimensions: $m \times n$. We have $M_T \in \mathbb{R}^{m\times n}$ since we used n "probe vectors," and since the outputs of T are m-dimensional column vectors.

To help visualize this new "column" idea, let's analyze what happens when we compute the product $M_T\hat{e}_2$. The probe vector $\hat{e}_2 \equiv (0,1,0,\ldots,0)^\mathsf{T}$ will "select" only the second column from M_T, thus we'll obtain the correct output: $M_T\hat{e}_2 = T(\hat{e}_2)$. Similarly, applying M_T to other basis vectors selects the other columns of M_T.

Any input vector can be written as a linear combination of the standard basis vectors $\vec{v} = v_1\hat{e}_1 + v_2\hat{e}_2 + \cdots + v_n\hat{e}_n$. Therefore, by linearity, we can compute the output $T(\vec{v})$ as follows:

$$\begin{aligned} T(\vec{v}) &= v_1T(\hat{e}_1) + v_2T(\hat{e}_2) + \cdots + v_nT(\hat{e}_n) \\ &= v_1\begin{bmatrix} | \\ T(\hat{e}_1) \\ | \end{bmatrix} + v_2\begin{bmatrix} | \\ T(\hat{e}_2) \\ | \end{bmatrix} + \cdots + v_n\begin{bmatrix} | \\ T(\hat{e}_n) \\ | \end{bmatrix} \\ &= \begin{bmatrix} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \ldots & T(\hat{e}_n) \\ | & | & & | \end{bmatrix}\begin{bmatrix} | \\ \vec{v} \\ | \end{bmatrix} \\ &= M_T[\vec{v}], \end{aligned}$$

where $[\vec{v}] = (v_1, v_2, \ldots, v_n)^\mathsf{T}$ is the coefficients vector of $\vec{v}$, represented as a column vector.

Input and output spaces

We can identify correspondences between the properties of a linear transformation T and the properties of a matrix M_T that implements T.

The outputs of the linear transformation T consist of all possible linear combinations of the columns of the matrix M_T. Thus, we can identify the *image space* of the linear transformation T with the *column space* of the matrix M_T:

$$\text{Im}(T) = \{\vec{w} \in W \mid \vec{w} = T(\vec{v}), \text{ for some } \vec{v} \in V\} = \mathcal{C}(M_T).$$

There is also an equivalence between the kernel of the linear transformation T and the null space of the matrix M_T:

$$\text{Ker}(T) \equiv \{\vec{v} \in \mathbb{R}^n | T(\vec{v}) = \vec{0}\} = \{\vec{v} \in \mathbb{R}^n | M_T\vec{v} = \vec{0}\} \equiv \mathcal{N}(M_T).$$

The null space of a matrix $\mathcal{N}(M_T)$ consists of all vectors that are orthogonal to the rows of the matrix M_T. The vectors in the null space of M_T have a zero dot product with each of the rows of M_T. This orthogonality can also be phrased in the opposite direction. Any vector in the row space $\mathcal{R}(M_T)$ of the matrix is orthogonal to the null space $\mathcal{N}(M_T)$ of the matrix.

These observations allow us to decompose the domain of the transformation T as the *orthogonal sum* of the row space and the null space of the matrix M_T:

$$\mathbb{R}^n \;=\; \mathcal{R}(M_T) \oplus \mathcal{N}(M_T).$$

This split implies the *conservation of dimensions* formula:

$$\dim(\mathbb{R}^n) = n = \dim(\mathcal{R}(M_T)) + \dim(\mathcal{N}(M_T)).$$

The sum of the dimensions of the row space and the null space of a matrix M_T is equal to the total dimensions of the input space.

We can summarize everything we know about the input-output relationship of the linear transformation T as follows:

$$T\colon \mathcal{R}(M_T) \to \mathcal{C}(M_T), \qquad T\colon \mathcal{N}(M_T) \to \{\vec{0}\}.$$

Input vectors $\vec{v} \in \mathcal{R}(M_T)$ are mapped to output vectors $\vec{w} \in \mathcal{C}(M_T)$ in a one-to-one correspondence. Input vectors $\vec{v} \in \mathcal{N}(M_T)$ are mapped to the zero vector $\vec{0} \in \mathbb{R}^m$.

Composition of linear transformations

The consecutive application of two linear transformations T and S on an input vector $\vec{v}$ corresponds to the following matrix product:

$$S \circ T(\vec{v}) \;=\; S(T(\vec{v})) \;=\; M_S M_T \vec{v}.$$

The matrix M_T "touches" the vector first, followed by the matrix M_S.

For this composition to be well-defined, the dimension of the output space of T must be the same as the dimension of the input space of S. In terms of the matrices, this requirement corresponds to the condition that the *inner dimension* in the matrix product $M_S M_T$ must be the same.

Importance of the choice of bases

Above, we assumed the standard basis was used both for inputs and outputs of the linear transformation. Thus, we obtained the coefficients in the matrix M_T *with respect to* the standard basis.

In particular, we assumed that the outputs of T were given as column vectors in terms of the standard basis for $\mathbb{R}^m$. If the outputs were given in some other basis B_W, the coefficients of the matrix M_T would have been *with respect to* B_W.

Due to the dependence of matrix coefficients on the basis used, **it's wrong to say that a linear transformation *is* a matrix**. Indeed, the same linear transformation T will correspond to different

matrices if different bases are used. We say the linear transformation T corresponds to a matrix M for a *given* choice of input and output bases. We write ${}_{B_W}[M_T]_{B_V}$ to show the coefficients of the matrix M_T depend on the choice of left and right bases. Recall we can also use the basis-in-a-subscript notation for vectors. For example, writing $(v_x, v_y, v_z)_{B_s}$ shows the coefficients v_x, v_y, and v_z are expressed in terms of the standard basis $B_s \equiv \{\hat{\imath}, \hat{\jmath}, \hat{k}\}$.

The choice of basis is an important technical detail: be aware of it, but don't worry about it too much. Unless otherwise specified, assume the standard basis is used for specifying vectors and matrices. When you see the product $A\vec{v}$, it means ${}_{B_s}[A]_{B_s}\,[\vec{v}]_{B_s}$. The only time you really need to pay attention to the choice of bases is when performing *change-of-basis* transformations, which we'll discuss in Section 6.3.

Invertible transformations

We'll now revisit the properties of invertible matrices and connect them to the notion of invertible transformations. Think of multiplication by a matrix M as "doing" something to vectors, and multiplication by M^{-1} as doing the opposite thing, restoring the original vector:

$$M^{-1}M\vec{v} = \vec{v}.$$

For example, the matrix

$$M = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

corresponds to a stretching of space by a factor of 2 in the x-direction, while the y-direction remains unchanged. The inverse transformation corresponds to a shrinkage by a factor of 2 in the x-direction:

$$M^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}.$$

In general, it's hard to see exactly what the matrix M does, since it performs some arbitrary linear combination of the coefficients of the input vector.

If M is an invertible matrix, we can start from any output vector $\vec{w} = M\vec{v}$, and go back to find the input $\vec{v}$ that produced the output $\vec{w}$. We do this by multiplying $\vec{w}$ by the inverse: $M^{-1}\vec{w} = M^{-1}M\vec{v} = \vec{v}$.

A linear transformation T is *invertible* if there exist an inverse transformation T^{-1} such that $T^{-1}(T(\vec{v})) = \vec{v}$. By the correspondence $\vec{w} = T(\vec{v}) \;\Leftrightarrow\; \vec{w} = M_T\vec{v}$, **we can identify the class of invertible linear transformations with the class of invertible matrices**.

Affine transformations

An *affine transformation* is a function $A : \mathbb{R}^n \to \mathbb{R}^m$ that is the combination of a linear transformation T followed by a *translation* by a fixed vector $\vec{b}$:

$$\vec{y} = A(\vec{x}) = T(\vec{x}) + \vec{b}.$$

By the $T \Leftrightarrow M_T$ equivalence we can write the formula for an affine transformation as

$$\vec{y} = A(\vec{x}) = M_T\vec{x} + \vec{b}.$$

To obtain the output, $\vec{y}$, apply the linear transformation T (the matrix-vector product $M_T\vec{x}$), then add the vector $\vec{b}$. This is the vector generalization of a single-variable *affine function* $y = f(x) = mx + b$.

Discussion

The most general linear transformation

In this section we learned that a linear transformation *can* be represented as matrix multiplication. Are there other ways to represent linear transformations? To study this question, let's analyze, from first principles, the most general form a linear transformation $T\colon V \to W$ can take. We'll use $V = \mathbb{R}^3$ and $W = \mathbb{R}^2$ to keep things simple.

First consider the coefficient w_1 of the output vector $\vec{w} = T(\vec{v})$ when the input vector is $\vec{v} \in \mathbb{R}^3$. The fact that T is linear, means that w_1 can be an arbitrary *mixture* of the input vector coefficients v_1, v_2, v_3:

$$w_1 = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3.$$

Similarly, the coefficient w_2 must be some arbitrary linear combination of the input coefficients $w_2 = \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3$. Thus, the most general linear transformation $T\colon \mathbb{R}^3 \to \mathbb{R}^2$ can be written as

$$\begin{aligned} w_1 &= \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3, \\ w_2 &= \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3. \end{aligned}$$

This is precisely the kind of expression that can be obtained as a matrix product:

$$T(\vec{v}) = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = M_T\vec{v}.$$

Indeed, the matrix product is defined as rows-dot-columns because it allows us to easily describe linear transformations.

Links

[Examples of linear transformations from Wikibooks]
`http://wikibooks.org/wiki/Linear_Algebra/Linear_Transformations`

Exercises

E6.1 Consider the transformation $T(x, y, z) = (y + z, x + z, x + y)$. Find the domain, codomain, kernel, and image space of T. Is T injective, surjective, or bijective?

6.2 Finding matrix representations

Every linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ can be represented as a matrix $M_T \in \mathbb{R}^{m \times n}$. Suppose you're given the following description of a linear transformation: "T is the counterclockwise rotation of all points in the xy-plane by 30°," and you want to find the matrix M_T that corresponds to this transformation.

Do you know how to find the matrix representation of T? This section describes a simple and intuitive "probing procedure" for finding matrix representations. Don't worry; no alien technology is involved, and we won't be probing any humans—only linear transformations! As you read, try to bridge your understanding between the general *specification* of a transformation $T(\vec{v})$ and its specific *implementation* as a matrix-vector product $M_T\vec{v}$. We'll use the "probing procedure" to study various linear transformations and derive their matrix representations.

Once we find the matrix representation of a given transformation, we can efficiently apply that transformation to many vectors. This is exactly how computers carry out linear transformations. For example, a black-and-white image file can be represented as a long list that contains the coordinates of the image's black pixels: $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_\ell\}$. The image is obtained by starting with a white background and drawing a black pixel in each of the locations $\vec{x}_i$ on the screen.[1] To rotate the image, we can process the list of pixels using the matrix-vector product $\vec{y}_i = M_T\vec{x}_i$, where M_T is the matrix representation of the desired rotation. The transformed list of pixels $\{\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_\ell\}$ corresponds to a rotated version of the image. This is essentially the effect of using the "rotate tool" in an image editing program—the computer multiplies the image by a rotation matrix.

[1] Location on a computer screen is denoted using pixel coordinates $\vec{x}_i = (h_i, v_i)$. The number h_i describes a horizontal distance measured in pixels from the left edge of the image, and v_i measures the vertical distance from the top of the image.

Concepts

The previous section covered linear transformations and their matrix representations:

- $T : \mathbb{R}^n \to \mathbb{R}^m$: a linear transformation that takes inputs in $\mathbb{R}^n$ and produces outputs in $\mathbb{R}^m$
- $M_T \in \mathbb{R}^{m \times n}$: the matrix representation of T

The action of the linear transformation T is equivalent to a multiplication by the matrix M_T:

$$\vec{w} = T(\vec{v}) \qquad \Leftrightarrow \qquad \vec{w} = M_T \vec{v}.$$

Theory

To find the matrix representation of the transformation $T \colon \mathbb{R}^n \to \mathbb{R}^m$, it is sufficient to "probe" T with the n vectors of the standard basis for the input space $\mathbb{R}^n$:

$$\hat{e}_1 \equiv \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \hat{e}_2 \equiv \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad \hat{e}_n \equiv \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The matrix M_T that corresponds to the action of T on the standard basis is

$$M_T = \begin{bmatrix} | & | & & | \\ T(\vec{e}_1) & T(\vec{e}_2) & \ldots & T(\vec{e}_n) \\ | & | & & | \end{bmatrix}.$$

This is an $m \times n$ matrix whose columns are the outputs of T for the n probe inputs.

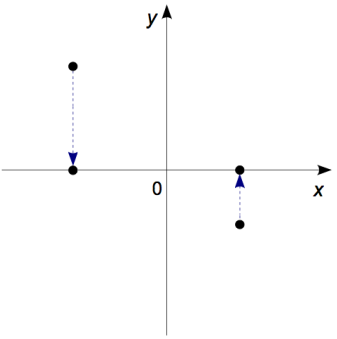
The remainder of this section illustrates the "probing procedure" for finding matrix representations of linear transformations through several examples.

Projections

We'll start with a class of linear transformations you're already familiar with: *projections*. I hope you still remember what you learned in Section 5.2 (page 189).

X projection

The projection onto the x-axis is denoted Π_x. The projection Π_x acts on any vector or point by leaving the x-coordinate unchanged and setting the y-coordinate to zero.



Let's analyze how the projection Π_x transforms the two vectors of the standard basis:

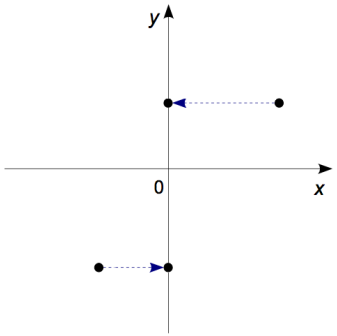
$$\Pi_x\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}, \quad \Pi_x\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}.$$

The matrix representation of Π_x is therefore given by:

$$M_{\Pi_x} = \begin{bmatrix}\Pi_x\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & \Pi_x\left(\begin{bmatrix}0\\1\end{bmatrix}\right)\end{bmatrix} = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}.$$

Y projection

Similar to Π_x, Π_y is defined as the projection onto the y-axis. Can you guess what the matrix for the projection onto the y-axis will look like?



Use the standard approach to find the matrix representation of Π_y:

$$M_{\Pi_y} = \begin{bmatrix}\Pi_y\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & \Pi_y\left(\begin{bmatrix}0\\1\end{bmatrix}\right)\end{bmatrix} = \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}.$$

We can verify that the matrices M_{Π_x} and M_{Π_y} do indeed select the appropriate coordinate from a general input vector $\vec{v} = (v_x, v_y)^\mathsf{T}$:

$$\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}v_x\\v_y\end{bmatrix} = \begin{bmatrix}v_x\\0\end{bmatrix}, \qquad \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}v_x\\v_y\end{bmatrix} = \begin{bmatrix}0\\v_y\end{bmatrix}.$$

Projection onto a vector

Recall that the general formula for the projection of a vector $\vec{v}$ onto another vector $\vec{a}$ is obtained as:

$$\Pi_{\vec{a}}(\vec{v}) = \left(\frac{\vec{a}\cdot\vec{v}}{\|\vec{a}\|^2}\right)\vec{a}.$$

To find the matrix representation of a projection onto an arbitrary direction $\vec{a}$, we compute

$$M_{\Pi_{\vec{a}}} = \begin{bmatrix}\Pi_{\vec{a}}\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & \Pi_{\vec{a}}\left(\begin{bmatrix}0\\1\end{bmatrix}\right)\end{bmatrix}.$$

Projection onto a plane

We can also compute the projection of the vector $\vec{v} \in \mathbb{R}^3$ onto the plane $P: \ \vec{n} \cdot \vec{x} = n_x x + n_y y + n_z z = 0$ as follows:

$$\Pi_P(\vec{v}) = \vec{v} - \Pi_{\vec{n}}(\vec{v}).$$

How should we interpret the above formula? First compute the part of the vector $\vec{v}$ that is perpendicular to the plane (in the $\vec{n}$ direction), then subtract this part from $\vec{v}$ to obtain the part that lies in the plane.

To obtain the matrix representation of Π_P, calculate what it does to the standard basis $\hat{\imath} \equiv \hat{e}_1 \equiv (1,0,0)^\mathsf{T}$, $\hat{\jmath} \equiv \hat{e}_2 \equiv (0,1,0)^\mathsf{T}$, and $\hat{k} \equiv \hat{e}_3 \equiv (0,0,1)^\mathsf{T}$.

Projections as outer products

We can obtain the projection matrix onto any unit vector by computing the *outer product* of the vector with itself. As an example, we'll find the matrix for the projection onto the x-axis $\Pi_x(\vec{v}) = (\vec{v} \cdot \hat{\imath})\hat{\imath}$. Recall the *inner product* (dot product) between two column vectors $\vec{u}$ and $\vec{v}$ is equivalent to the matrix product $\vec{u}^\mathsf{T}\vec{v}$, while their *outer product* is given by the matrix product $\vec{u}\vec{v}^\mathsf{T}$. The inner product is a product between a $1 \times n$ matrix and a $n \times 1$ matrix, whose result is a 1×1 matrix—a single number. The outer product corresponds to an $n \times 1$ matrix times a $1 \times n$ matrix, making the answer an $n \times n$ matrix. The projection matrix corresponding to Π_x is $M_{\Pi_x} = \hat{\imath}\hat{\imath}^\mathsf{T} \in \mathbb{R}^{n \times n}$.

Where did that equation come from? To derive the equation, we ① use the commutative law of scalar multiplication $\alpha\vec{v} = \vec{v}\alpha$, ② rewrite the dot product formula as a matrix product, and ③ use the *associative* law of matrix multiplication $A(BC) = (AB)C$. Check it:

$$\begin{aligned}
\Pi_x(\vec{v}) = (\hat{\imath} \cdot \vec{v})\,\hat{\imath} \overset{①}{=} \hat{\imath}(\hat{\imath} \cdot \vec{v}) \overset{②}{=} \hat{\imath}(\hat{\imath}^\mathsf{T}\vec{v}) &= \begin{bmatrix}1\\0\end{bmatrix}\left(\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}v_x\\v_y\end{bmatrix}\right) \\
\overset{③}{=} \left(\hat{\imath}\hat{\imath}^\mathsf{T}\right)\vec{v} &= \left(\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}\right)\begin{bmatrix}v_x\\v_y\end{bmatrix} \\
= (M)\,\vec{v} &= \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}v_x\\v_y\end{bmatrix} = \begin{bmatrix}v_x\\0\end{bmatrix}.
\end{aligned}$$

The outer product $M \equiv \hat{\imath}\hat{\imath}^\mathsf{T}$ corresponds to the projection matrix M_{Π_x} we're looking for.

More generally, we obtain the projection matrix onto a line with direction vector $\vec{a}$ by constructing the unit vector $\hat{a}$, and then calculating the outer product of $\hat{a}$ with itself:

$$\hat{a} \equiv \frac{\vec{a}}{\|\vec{a}\|}, \qquad M_{\Pi_{\vec{a}}} = \hat{a}\hat{a}^\mathsf{T}.$$

Example Find the projection matrix $M_d \in \mathbb{R}^{2\times 2}$ for the projection Π_d onto the line with equation $y = x$ (45° diagonal line).

The line with equation $y = x$ corresponds to the parametric equation $\{(x, y) \in \mathbb{R}^2 \mid (x, y) = (0, 0) + t(1, 1),\ t \in \mathbb{R}\}$, so the direction vector for this line is $\vec{a} = (1, 1)$. We want to find the matrix that corresponds to the linear transformation $\Pi_d(\vec{v}) = \left(\frac{\vec{v}\cdot(1,1)}{2}\right)(1,1)^\mathsf{T}$.

The projection matrix onto $\vec{a} = (1, 1)$ is computed using the outer product approach. First, compute a normalized direction vector $\hat{a} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$; then compute M_d using the outer product:

$$M_d = \hat{a}\hat{a}^\mathsf{T} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Usually, the idea of representing projections as outer products is not covered in a first linear algebra course, so don't worry about outer products appearing on the exam. The purpose of introducing you to the equivalence between projections onto $\hat{a}$ and the outer product $\hat{a}\hat{a}^\mathsf{T}$ is to illuminate this interesting connection between vectors and matrices. This connection plays a fundamental role in quantum mechanics, where projections in different directions are frequently used.

If you're asked a matrix representation question on an exam, keep things simple and stick to the "probing with the standard basis" approach, which gives the same answer as the one computed using the outer product:

$$M_d = \begin{bmatrix} \Pi_d\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) & \Pi_d\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \end{bmatrix} = \begin{bmatrix} \left(\frac{\hat{\imath}\cdot\vec{a}}{\|\vec{a}\|^2}\right)\vec{a} & \left(\frac{\hat{\jmath}\cdot\vec{a}}{\|\vec{a}\|^2}\right)\vec{a} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Projections are idempotent

A projection matrix M_Π satisfies $M_\Pi M_\Pi = M_\Pi$. This is one of the defining properties of projections. The technical term for this is *idempotence*, meaning the operation can be applied multiple times without changing the result beyond the initial application.

Subspaces

A projection acts differently on different sets of input vectors. While some input vectors remain *unchanged*, some input vectors are *killed*. This is murder! Well, murder in a *mathematical* sense, which means multiplication by zero.

Let Π_S be the projection onto the space S, and $S^\perp$ be the *orthogonal space* to S defined by $S^\perp \equiv \{\, \vec{w} \in \mathbb{R}^n \mid \vec{w} \cdot \vec{s} = 0,\ \forall \vec{s} \in S \,\}$. The

action of Π_S is completely different for the vectors from S and $S^\perp$. All vectors $\vec{v}$ in S remain unchanged:

$$\Pi_S(\vec{v}) = \vec{v},$$

whereas vectors $\vec{w}$ in $S^\perp$ are *killed*:

$$\Pi_S(\vec{w}) = 0\vec{w} = \vec{0}.$$

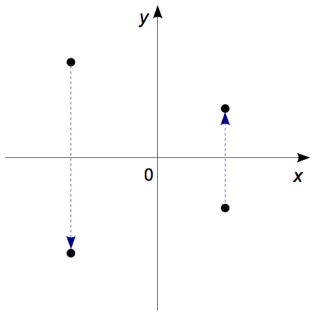
The action of Π_S on any vector from $S^\perp$ is equivalent to multiplication by zero. This is why $S^\perp$ is called the *null space* of M_{Π_S}.

Reflections

We'll now compute matrix representations for simple *reflection* transformations.

X reflection

Reflection through the x-axis leaves the x-coordinate unchanged and flips the sign of the y-coordinate. The figure on the right illustrates the effect of reflecting two points through the x-axis.



Using the usual probing procedure, we obtain the matrix that corresponds to this linear transformation:

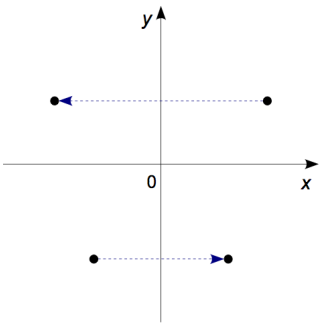
$$M_{R_x} = \begin{bmatrix} R_x\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & R_x\left(\begin{bmatrix}0\\1\end{bmatrix}\right) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

This matrix sends $(x, y)^\mathsf{T}$ to $(x, -y)^\mathsf{T}$ as required.

Y reflection

The matrix associated with R_y, the reflection through the y-axis, is given by:

$$M_{R_y} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$



The numbers in the above matrix tell us to change the sign of the x-coordinate of the input and leave its y-coordinate unchanged. In other words, every point that starts on the left of the y-axis moves to the right of

the y-axis, and every point that starts on the right of the y-axis moves to the left.

Do you see the power and simplicity of the probing procedure for finding matrix representations? In the first column, enter what you want to happen to the $\hat{e}_1$ vector; in the second column, enter what you want to happen to the $\hat{e}_2$ vector, and voila!

Diagonal reflection

Suppose we want to find the formula for the reflection through the line $y = x$. We'll call this reflection R_d. This time, dear readers, it's up to you to draw the diagram. In words, R_d is the transformation that makes x and y "swap places."

Based on this notion of swapping places, the matrix for R_d is

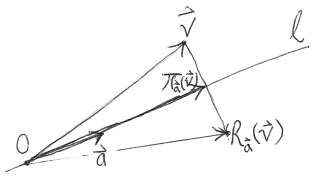
$$M_{R_d} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Alternatively, the usual "probing procedure" will lead us to the same result.

Now I must point out an important property common to all reflections. The effect of a reflection is described by one of two possible actions: some points remain unchanged by the reflection, while other points flip into their exact negatives. For example, the *invariant* points under R_y are the points that lie on the y-axis—that is, the multiples of $(0, 1)^\mathsf{T}$. The points that become their *exact negatives* are those with only an x-component—the multiples of $(1, 0)^\mathsf{T}$. The action of R_y on all other points can be obtained as a linear combination of the actions "leave unchanged" and "multiply by -1." We'll extent this line of reasoning further at the end of this section, and again in Section 7.1.

Reflections through lines and planes

What about finding reflections through an arbitrary line? Consider the line with parametric equation $\ell : \{\vec{0} + t\vec{a}, t \in \mathbb{R}\}$. We can find a formula for the reflection through ℓ in terms of the projection formula we obtained earlier:

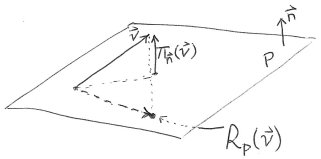


$$R_{\vec{a}}(\vec{v}) = 2\Pi_{\vec{a}}(\vec{v}) - \vec{v}.$$

Consider how we arrive at this formula. First, compute the projection of $\vec{v}$ onto the line $\Pi_{\vec{a}}(\vec{v})$. Then take two steps in that direction and subtract $\vec{v}$ once. Annotate the figure with a pencil so you can visualize that the formula works.

Similarly, we can derive an expression for the reflection through an arbitrary plane $P: \ \vec{n} \cdot \vec{x} = 0$:

$$R_P(\vec{v}) = 2\Pi_P(\vec{v}) - \vec{v} = \vec{v} - 2\Pi_{\vec{n}}(\vec{v}).$$



The first form of the formula uses a reasoning similar to the formula for the reflection through a line.

The second form of the formula can be understood as computing the shortest vector from the plane to $\vec{v}$, subtracting that vector once from $\vec{v}$ to reach a point in the plane, and subtracting it a second time to move to the point $R_P(\vec{v})$ on the other side of the plane.

Rotations

We'll now find the matrix representations for *rotation* transformations. The rotation in the counterclockwise by the angle θ is denoted R_θ. Figure 6.5 illustrates the action of the rotation R_θ: the point A is rotated around the origin to become the point B.

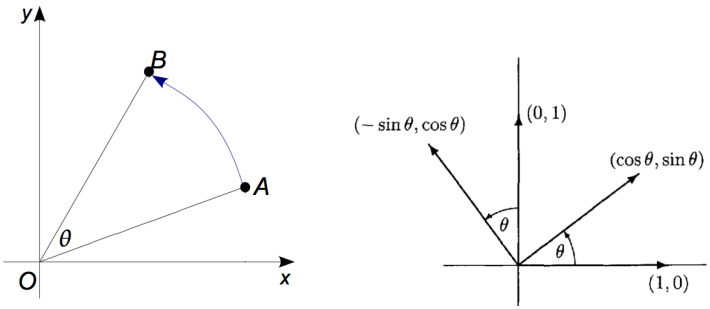


Figure 6.5: The linear transformation R_θ rotates every point in the plane by the angle θ in the counterclockwise direction. Note the effect of R_θ on the basis vectors $(1, 0)$ and $(0, 1)$.

To find the matrix representation of R_θ, probe it with the standard basis as usual:

$$M_{R_\theta} = \begin{bmatrix} R_\theta\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & R_\theta\left(\begin{bmatrix}0\\1\end{bmatrix}\right) \end{bmatrix}.$$

To compute the values in the first column, observe that R_θ rotates the vector $(1, 0)^T = 1\angle 0$ to the vector $1\angle\theta = (\cos\theta, \sin\theta)^T$. The second

input $\hat{e}_2 = (0,1)^T = 1\angle\frac{\pi}{2}$ is rotated to $1\angle(\frac{\pi}{2}+\theta) = (-\sin\theta, \cos\theta)^T$. Therefore, the matrix for R_θ is

$$M_{R_\theta} = \begin{bmatrix} | & | \\ 1\angle\theta & 1\angle(\frac{\pi}{2}+\theta) \\ | & | \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Finding the matrix representation of a linear transformation is like a colouring-book activity for mathematicians. Filling in the columns is just like colouring inside the lines—nothing complicated.

Inverses

Can you determine the inverse matrix of M_{R_θ}? You could use the formula for finding the inverse of a 2×2 matrix, *or* you could use the $[\ A \mid I\ ]$-and-RREF algorithm for finding the inverse; but using these approaches would be *waaaaay* too much work. Try to guess the matrix representation of the inverse without doing any calculations. If R_θ rotates points by $+\theta$, can you tell me what the inverse operation does? I'll leave a blank line here to give you some time to think....

Think you have it? The inverse operation of R_θ is $R_{-\theta}$, a rotation by $-\theta$, which corresponds to the matrix

$$M_{R_{-\theta}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Recall that cos is an even function, so $\cos(-\theta) = \cos(\theta)$, while sin is an odd function, so $\sin(-\theta) = -\sin(\theta)$.

For any vector $\vec{v} \in \mathbb{R}^2$ we have $R_{-\theta}\left(R_\theta(\vec{v})\right) = \vec{v} = R_\theta\left(R_{-\theta}(\vec{v})\right)$, or in terms of matrices:

$$M_{R_{-\theta}} M_{R_\theta} = \mathbb{1} = M_{R_\theta} M_{R_{-\theta}}.$$

Cool, right? This is what *representation* really means: the abstract notion of composition of linear transformations is *represented* as a matrix product.

Here's another quiz question: what is the inverse operation of the reflection through the x-axis R_x? The "undo" action for R_x is to apply R_x again. We say R_x is a self-inverse operation.

What is the inverse matrix of a projection Π_S? Good luck finding that one—it's was a trick question. The projection Π_S sends all input vectors from the subspace $S^\perp$ to the zero vector. Projections are inherently many-to-one transformations and therefore not invertible.

Nonstandard-basis probing

At this point you should feel confident facing any linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$, and probing it with the standard basis to find its matrix representation $M_T \in \mathbb{R}^{2\times 2}$. But what if you're not allowed to probe T with the standard basis? What if you must find the matrix of the transformation given the outputs of T for some other basis $\{\vec{v}_1 \equiv (v_{1x}, v_{1y})^\mathsf{T}, \vec{v}_2 \equiv (v_{2x}, v_{2y})^\mathsf{T}\}$:

$$T\left(\begin{bmatrix} v_{1x} \\ v_{1y} \end{bmatrix}\right) = \begin{bmatrix} t_{1x} \\ t_{1y} \end{bmatrix}, \qquad T\left(\begin{bmatrix} v_{2x} \\ v_{2y} \end{bmatrix}\right) = \begin{bmatrix} t_{2x} \\ t_{2y} \end{bmatrix}.$$

Let's test this idea. We're given the information $v_{1x}, v_{1y}, t_{1x}, t_{1y}$, and $v_{2x}, v_{2y}, t_{2x}, t_{2y}$, and must find the matrix representation of T with respect to the standard basis.

Because the vectors $\vec{v}_1$ and $\vec{v}_2$ form a basis, we can reconstruct the information about the matrix M_T from the input-output information given. We're looking for four unknowns—m_{11}, m_{12}, m_{21}, and m_{22}—that form the matrix representation of T:

$$M_T = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}.$$

We can write four equations with the input-output information provided:

$$\begin{aligned} m_{11}v_{1x} + m_{12}v_{1y} &= t_{1x}, \\ m_{21}v_{1x} + m_{22}v_{1y} &= t_{1y}, \\ m_{11}v_{2x} + m_{12}v_{2y} &= t_{2x}, \\ m_{21}v_{2x} + m_{22}v_{2y} &= t_{2y}. \end{aligned}$$

Since there are four equations and four unknowns, we can solve for the coefficients m_{11}, m_{12}, m_{21}, and m_{22}.

This system of equations differs from ones we've seen before, so we'll examine it in detail. Think of the entries of M_T as a 4×1 vector of unknowns $\vec{m} = (m_{11}, m_{12}, m_{21}, m_{22})^\mathsf{T}$. We can rewrite the four equations as a matrix equation:

$$A\vec{m} = \vec{t} \qquad \Leftrightarrow \qquad \begin{bmatrix} v_{1x} & v_{1y} & 0 & 0 \\ 0 & 0 & v_{1x} & v_{1y} \\ v_{2x} & v_{2y} & 0 & 0 \\ 0 & 0 & v_{2x} & v_{2y} \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{21} \\ m_{22} \end{bmatrix} = \begin{bmatrix} t_{1x} \\ t_{1y} \\ t_{2x} \\ t_{2y} \end{bmatrix}.$$

Next, solve for $\vec{m}$ by computing $\vec{m} = A^{-1}\vec{t}$.

Finding the matrix representation by probing with a nonstandard basis is more work than probing with the standard basis, but it's totally doable.

Eigenspaces

Probing the transformation T with *any* basis for the input space gives sufficient information to determine its matrix representation. We're free to choose the "probing basis," so how do we decide which basis to use? The standard basis is good for computing the matrix representation, but perhaps there's a basis that allows us to simplify the abstract description of T; a so-called *natural* basis for probing each transformation.

Indeed, such a basis exists. Many linear transformations have a basis $\{\vec{e}_{\lambda_1}, \vec{e}_{\lambda_2}, \ldots\}$ such that the action of T on the basis vector $\vec{e}_{\lambda_i}$ is equivalent to the scaling of $\vec{e}_{\lambda_i}$ by the constant λ_i:

$$T(\vec{e}_{\lambda_i}) = \lambda_i \vec{e}_{\lambda_i}.$$

Recall how projections leave some vectors unchanged (multiply by $\lambda = 1$) and send other vectors to the zero vector (multiply by $\lambda = 0$). These subspaces of the input space are specific to each transformation, and are called the *eigenspaces* (from the German "own spaces") of the transformation T.

Consider the reflection R_x and its two eigenspaces:

- The space of vectors that remain unchanged (the eigenspace corresponding to $\lambda_1 = 1$) is spanned by the vector $(1, 0)$:

$$R_x\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = 1\begin{bmatrix}1\\0\end{bmatrix}.$$

- The space of vectors that become the exact negatives of themselves (corresponding to $\lambda_2 = -1$) is spanned by $(0, 1)$:

$$R_x\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = -1\begin{bmatrix}0\\1\end{bmatrix}.$$

From a theoretical point of view, describing the action of a liner transformation T in its natural basis is the best way to understand it. For each of the *eigenvectors* in the various eigenspaces of T, the action of T is a simple scalar multiplication. We defer the detailed discussion on *eigenvalues* and *eigenvectors* until the next chapter.

Links

[Rotation operation as the composition of three shear operations]
`http://datagenetics.com/blog/august32013/index.html`

Exercises

6.3 Change of basis for matrices

Every linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ can be represented as a matrix $M_T \in \mathbb{R}^{m \times n}$. The coefficients of the matrix M_T depend on the basis used to describe the input and output spaces. Note, this dependence of matrix coefficients on the basis is directly analogous to the dependence of vector coefficients on the basis.

In this section we'll learn how the choice of basis affects the coefficients of matrix representations, and discuss how to carry out the change-of-basis operation for matrices.

Concepts

You should already be familiar with the concepts of vector spaces, bases, vector coefficients with respect to different bases, and the change-of-basis transformation:

- V: an n-dimensional vector space
- $\vec{v}$: a vector in V
- $B = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$: an orthonormal basis for V
- $[\vec{v}]_B = (v_1, v_2, \ldots, v_n)_B$: the vector $\vec{v}$ expressed in the basis B
- $B' = \{\hat{e}'_1, \hat{e}'_2, \ldots, \hat{e}'_n\}$: another orthonormal basis for V
- $[\vec{v}]_{B'} = (v'_1, v'_2, \ldots, v'_n)_{B'}$: the vector $\vec{v}$ expressed in the basis B'
- ${}_{B'}[\mathbb{1}]_B$: the change-of-basis matrix that converts from B coordinates to B' coordinates: $[\vec{v}]_{B'} = {}_{B'}[\mathbb{1}]_B \, [\vec{v}]_B$
- ${}_{B}[\mathbb{1}]_{B'}$: the inverse change-of-basis matrix $[\vec{v}]_B = {}_{B}[\mathbb{1}]_{B'} \, [\vec{v}]_{B'}$ (note that ${}_{B}[\mathbb{1}]_{B'} = ({}_{B'}[\mathbb{1}]_B)^{-1}$)

We'll use the following concepts when describing a linear transformation $T : V \to W$:

- $B_V = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$: a basis for the input vector space V
- $B_W = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_m\}$: a basis for the output vector space W
- ${}_{B_W}[M_T]_{B_V} \in \mathbb{R}^{m \times n}$: a matrix representation of the linear transformation T with respect to the bases B_V and B_W:

$$\vec{w} = T(\vec{v}) \qquad \Leftrightarrow \qquad [\vec{w}]_{B_W} = {}_{B_W}[M_T]_{B_V} \, [\vec{v}]_{B_V}.$$

By far, the most commonly used basis in linear algebra is the standard basis $\hat{e}_1 = (1, 0, 0, \ldots)^T$, $\hat{e}_2 = (0, 1, 0, \ldots)^T$, etc. It is therefore customary to denote the matrix representation of a linear transformation T simply as M_T, without an explicit reference to the input

and output bases. This simplified notation causes much confusion and suffering when students later try to learn about change-of-basis operations.

In order to *really* understand the connection between linear transformations and their matrix representations, we need to have a little talk about bases and matrix coefficients. It's a little complicated, but the mental effort you invest is worth the overall understanding you'll gain. As the old Samurai saying goes, "Cry during training, laugh on the battlefield."

By the end of this section, you'll be able to handle any basis question your teacher may throw at you.

Matrix components

Every linear transformation T can be represented as a matrix M_T. Consider the linear transformation $T : V \to W$. Assume the input vector space V is n-dimensional and let $B_V = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ be a basis for V. Assume the output space W is m-dimensional and let $B_W = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_m\}$ be a basis for output space of T. The coefficients of the matrix $M_T \in \mathbb{R}^{m \times n}$ depend on the bases B_V and B_W. We'll now analyze this dependence in detail.

To compute the matrix representation of T with respect to the input basis $B_V = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$, we "probe" T with each of the vectors in the basis and record the outputs as the columns of a matrix:

$$[M_T]_{B_V} = \begin{bmatrix} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \ldots & T(\hat{e}_n) \\ | & | & & | \end{bmatrix}_{B_V}$$

The subscript B_V indicates the columns are built from outputs of the basis B_V. We can use the matrix $[M_T]_{B_V}$ to compute $T(\vec{v})$ for a vector $\vec{v}$ expressed in the basis B_V: $\vec{v} = (v_1, v_2, \ldots, v_n)^{\mathsf{T}}_{B_V}$. The matrix-vector product produces the correct linear combination of outputs:

$$\begin{aligned} [M_T]_{B_V}\,[\vec{v}]_{B_V} &= \begin{bmatrix} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \ldots & T(\hat{e}_n) \\ | & | & & | \end{bmatrix}_{B_V} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}_{B_V} \\ &= v_1 T(\hat{e}_1) + v_2 T(\hat{e}_2) + \cdots + v_n T(\hat{e}_n) \\ &= T(v_1 \hat{e}_1 + v_2 \hat{e}_2 + \cdots + v_n \hat{e}_n) \\ &= T(\vec{v}). \end{aligned}$$

So far we've been treating the outputs of T as abstract vectors $T(\hat{e}_j) \in W$. Like all vectors in the space W, each output of T can be expressed as a vector of coefficients with respect to the basis B_W. For example, the output $T(\hat{e}_1)$ can be expressed as

$$T(\hat{e}_1) = \begin{bmatrix} c_{11} \\ c_{21} \\ \vdots \\ c_{m1} \end{bmatrix}_{B_W} = c_{11}\vec{b}_1 + c_{21}\vec{b}_2 + \cdots + c_{m1}\vec{b}_m,$$

for some coefficients $c_{11}, c_{21}, \ldots, c_{m1}$. Similarly, the other output vectors $T(\hat{e}_j)$ can be expressed as coefficients with respect to the basis B_W, $T(\hat{e}_j) = (c_{1j}, c_{2j}, \ldots, c_{mj})^{\mathsf{T}}_{B_W}$.

We're now in a position to find the matrix representation ${}_{B_W}[M_T]_{B_V}$ of the linear transformation T, with respect to the input basis B_V and the output basis B_W:

$${}_{B_W}[M_T]_{B_V} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}_{B_W B_V} \in \mathbb{R}^{m \times n}.$$

The action of T on a vector $\vec{v}$ is the same as the product of ${}_{B_W}[M_T]_{B_V}$ and the vector of coefficients $[\vec{v}]_{B_V} = (v_1, v_2, \ldots, v_n)^{\mathsf{T}}_{B_V}$:

$$[T(\vec{v})]_{B_W} = {}_{B_W}[M_T]_{B_V}\,[\vec{v}]_{B_V}\,.$$

You may feel this example has stretched the limits of your attention span, but bear in mind, these nitty-gritty details hold the meaning of matrix coefficients. If you can see how the *positions* of the coefficients in the matrix encode the information about T and the choice of bases B_V and B_W, you're well on your way to getting it. The coefficient c_{ij} in the i^{th} row and j^{th} column in the matrix ${}_{B_W}[M_T]_{B_V}$ is the i^{th} component (with respect to B_W) of the output of T when the input is $\hat{e}_j$.

Verify that the matrix representation ${}_{B_W}[M_T]_{B_V}$ correctly predicts the output of T for the input $\vec{v} = 5\hat{e}_1 + 6\hat{e}_2 = (5, 6, 0, \ldots)^{\mathsf{T}}_{B_V}$. Using the linearity of T, we know the correct output is $T(\vec{v}) = T(5\hat{e}_1 + 6\hat{e}_2) = 5T(\hat{e}_1) + 6T(\hat{e}_2)$. We can verify that the matrix-vector product ${}_{B_W}[M_T]_{B_V}\,[\vec{v}]_{B_V}$ leads to the same answer:

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 0 \\ \vdots \end{bmatrix} = 5 \begin{bmatrix} c_{11} \\ c_{21} \\ \vdots \\ c_{m1} \end{bmatrix} + 6 \begin{bmatrix} c_{12} \\ c_{22} \\ \vdots \\ c_{m1} \end{bmatrix} = 5T(\hat{e}_1) + 6T(\hat{e}_2).$$

Change of basis for matrices

Given the matrix representation ${}_{B_W}[M_T]_{B_V}$ of the linear transformation $T : V \to W$, you're asked to find the matrix representation of T with respect to different bases B'_V and B'_W. This is the *change-of-basis* task for matrices.

We'll discuss the important special case where the input space and the output space of the linear transformation are the same. Let $T : V \to V$ be a linear transformation and let $B = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ and $B' = \{\hat{e}'_1, \hat{e}'_2, \ldots, \hat{e}'_n\}$ be two bases for the vector space V.

Recall the change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ that converts vectors from B coordinates to B' coordinates, and its inverse ${}_{B}[\mathbb{1}]_{B'}$, which converts vectors from B' coordinates to B coordinates:

$$[\vec{v}]_{B'} = {}_{B'}[\mathbb{1}]_B[\vec{v}]_B \qquad \text{and} \qquad [\vec{v}]_B = {}_{B}[\mathbb{1}]_{B'}[\vec{v}]_{B'} .$$

A clarification of notation is in order. The change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ is not equal to the identity matrix $\mathbb{1}_n$. However, the change-of-basis operation is *logically* equivalent to an identity transformation: the vector $\vec{v}$ doesn't change—only its coefficients. If you don't remember the change-of-basis operation for vectors, now's the time to flip back to Section 5.3 (page 198) and review before continuing.

Given the matrix representation ${}_{B}[M_T]_B$ of the linear transformation T with respect to B, we want to find the matrix ${}_{B'}[M_T]_{B'}$, which is the representation of T with respect to the basis B'. The computation is straightforward. Perform the change-of-basis operation on the input and output vectors:

$${}_{B'}[M_T]_{B'} = {}_{B'}[\mathbb{1}]_B \, {}_{B}[M_T]_B \, {}_{B}[\mathbb{1}]_{B'} .$$

This group of three matrices is interpreted as follows. Imagine an input vector $[\vec{v}]_{B'}$ multiplying the three matrices ${}_{B'}[\mathbb{1}]_B \, {}_{B}[M_T]_B \, {}_{B}[\mathbb{1}]_{B'}$ from the right. In the first step, ${}_{B}[\mathbb{1}]_{B'}$ converts the vector from the basis B' to the basis B so the matrix ${}_{B}[M_T]_B$ can be applied. In the last step, the matrix ${}_{B'}[\mathbb{1}]_B$ converts the output of ${}_{B}[M_T]_B$ to the basis B'. The combined effect of multiplying by this specific arrangement of three matrices is the same as applying T to the input vector $\vec{v}$:

$${}_{B'}[\mathbb{1}]_B \, {}_{B}[M_T]_B \, {}_{B}[\mathbb{1}]_{B'}[\vec{v}]_{B'} = [T(\vec{v})]_{B'} ,$$

which means

$${}_{B'}[M_T]_{B'} \equiv {}_{B'}[\mathbb{1}]_B \, {}_{B}[M_T]_B \, {}_{B}[\mathbb{1}]_{B'} .$$

This formula makes sense intuitively: to obtain a matrix with respect to a different basis, we must surround it by appropriate change-of-basis matrices.

Note the "touching dimensions" of the matrices are expressed with respect to the same basis. Indeed, we can think of the change-of-basis matrix as an "adaptor" we use to express vectors in a different basis. The change-of-basis operation for matrices requires two adaptors: one for the input space and one for the output space of the matrix.

Similarity transformation

It's interesting to note the abstract mathematical properties of the operation used above. Consider any matrix $N \in \mathbb{R}^{n \times n}$ and an invertible matrix $P \in \mathbb{R}^{n \times n}$. Define M to be the result when N is multiplied by P on the left and by the inverse P^{-1} on the right:

$$M = PNP^{-1}.$$

We say matrices N and M are related by a *similarity transformation*.

Since the matrix P is invertible, its columns form a basis for the vector space $\mathbb{R}^n$. Thus, we can interpret P as a *change-of-basis* matrix that converts the standard basis to the basis of the columns of P. The matrix P^{-1} corresponds to the inverse change-of-basis matrix. Using this interpretation, the matrix M corresponds to the *same* linear transformation as the matrix N, but is expressed with respect to the basis P.

Similarity transformations preserve certain properties of matrices:

- Trace: $\text{Tr}(M) = \text{Tr}(N)$
- Determinant: $\det(M) = \det(N)$
- Rank: $\text{rank}(M) = \text{rank}(N)$
- Eigenvalues: $\text{eig}(M) = \text{eig}(N)$

Together, the trace, the determinant, the rank, and the eigenvalues of a matrix are known as the invariant properties of the matrix because they don't depend on the choice of basis.

Exercises

E6.2 Suppose you're given the matrix representation of a linear transformation T with respect to the basis B': ${}_{B'}[M_T]_{B'}$. What formula describes the matrix representation of T with respect to the basis B?

6.4 Invertible matrix theorem

So far, we discussed systems of linear equations, matrices, vector spaces, and linear transformations. It's time to tie it all together! We'll now explore connections between these different contexts where

matrices are used. Originally, we explored how matrices can solve systems of linear equations. Later, we studied geometrical properties of matrices, including their row spaces, column spaces, and null spaces. In Chapter 6, we learned about the connection between matrices and linear transformations. In each of these domains, *invertible* matrices play a particularly important role. Lucky for us, there's a theorem that summarizes ten important facts about invertible matrices. One theorem; ten facts. Now that's a good deal!

Invertible matrix theorem: *For an $n \times n$ matrix A, the following statements are equivalent:*

(**1**) *A is invertible*

(**2**) *The equation $A\vec{x} = \vec{b}$ has exactly one solution for each $\vec{b} \in \mathbb{R}^n$*

(**3**) *The null space of A contains only the zero vector $\mathcal{N}(A) = \{\vec{0}\}$*

(**4**) *The equation $A\vec{x} = \vec{0}$ has only the trivial solution $\vec{x} = \vec{0}$*

(**5**) *The columns of A form a basis for $\mathbb{R}^n$:*
- *The columns of A are linearly independent*
- *The columns of A span $\mathbb{R}^n$; $\mathcal{C}(A) = \mathbb{R}^n$*

(**6**) *The rank of the matrix A is n*

(**7**) *The RREF of A is the $n \times n$ identity matrix $\mathbb{1}_m$*

(**8**) *The transpose matrix A^T is invertible*

(**9**) *The rows of A form a basis for $\mathbb{R}^n$:*
- *The rows of A are linearly independent*
- *The rows of A span $\mathbb{R}^n$; $\mathcal{R}(A) = \mathbb{R}^n$*

(**10**) *The determinant of A is nonzero* $\det(A) \neq 0$

These 10 statements are either all true or all false for a given matrix A. We can split the set of $n \times n$ matrices into two disjoint subsets: invertible matrices, for which all 10 statements are true, and non-invertible matrices, for which all statements are false.

Proof of the invertible matrix theorem

It's essential you understand the details of this proof; the reasoning that connects these 10 statements unites all the chunks of linear algebra we've discussed. If you don't consider yourself a proof person,

there's no excuse! Be sure to read and reread the proof, as it will help to solidify your understanding of the material covered thus far.

Proofs by contradiction

Since our arrival at the **invertible matrix theorem** marks an important step, we'll first quickly review some handy proof techniques, just to make sure everyone's ready. A *proof by contradiction* starts by assuming the opposite of the fact we want to prove, and after several derivation steps arrives at a contradiction—a mathematical inconsistency. Arriving at a contradiction implies our original premise is false, which means the fact we want to prove is true. Thus, to show that (A) implies (B)—denoted (A)⇒(B)—we can show that not-(B) implies not-(A). An example of a proof by contradiction is the proof of $\sqrt{2} \notin \mathbb{Q}$ (see page 75).

Review of definitions

To really make sure we're all on board before the train leaves the station, it's wise to review some definitions from previous chapters. The matrix $A \in \mathbb{R}^{n \times n}$ is *invertible* if there exists a matrix A^{-1} such that $AA^{-1} = \mathbb{1}_n = A^{-1}A$. The *null space* of A is the set of vectors that become the zero vector when multiplying A from the right: $\{\vec{v} \in \mathbb{R}^n \,|\, A\vec{v} = \vec{0}\}$. The *column space* $\mathcal{C}(A) \subseteq \mathbb{R}^n$ consists of all possible linear combinations of the columns of the matrix A. Similarly, the *row space* $\mathcal{R}(A) \subseteq \mathbb{R}^n$ consists of all possible linear combinations of the rows of A. The *rank* of a matrix, denoted $\text{rank}(A)$, is equal to the dimension of the row space and the column space $\text{rank}(A) = \dim(\mathcal{C}(A)) = \dim(\mathcal{R}(A))$. The rank of A is also equal to number of pivots (leading ones) in the reduced row echelon form of A. The *determinant* of A corresponds to the *scale factor* by which the linear transformation $T_A(\vec{x}) \equiv A\vec{x}$ transforms the n-dimensional volume of the hyper-cube in the input space when it maps it to the output space. If any of the columns of the matrix A are linearly dependent, the determinant $|A|$ will be zero. Phew!

Proof of the ***invertible matrix theorem****.* The moment has arrived: we'll prove the equivalence of the 10 statements in the theorem by showing a closed chain of implications between statements (**1**) through (**7**). We'll separately show the equivalences (**1**) ⇔ (**8**) ⇔ (**9**) and (**5**) ⇔ (**10**). Figure 6.6 shows an outline of the proof.

(1)⇒(2): Assume A is invertible so there exists an inverse matrix A^{-1} such that $A^{-1}A = \mathbb{1}_n$. Therefore, for all $\vec{b} \in \mathbb{R}^n$, the expression $\vec{x} = A^{-1}\vec{b}$ is a solution to $A\vec{x} = \vec{b}$. We must show the solution $\vec{x} = A^{-1}\vec{b}$

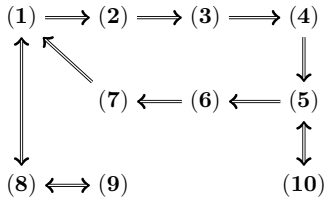


Figure 6.6: The chain of implications used to prove the **invertible matrix theorem**.

is the unique solution to this equation. Assume, for a contradiction, that a different solution $\vec{y} \neq \vec{x}$ exists, which also satisfies the equation $A\vec{y} = \vec{b}$. Multiplying both sides of the equation $A\vec{y} = \vec{b}$ by A^{-1}, we obtain $A^{-1}A\vec{y} = \vec{y} = A^{-1}\vec{b}$. We see that $\vec{y} = A^{-1}\vec{b} = \vec{x}$, which is contrary to our assumption that $\vec{y} \neq \vec{x}$. Thus, our assumption that a second solution $\vec{y} \neq \vec{x}$ exists is false, and the equation $A\vec{x} = \vec{b}$ has the unique solution $\vec{x} = A^{-1}\vec{b}$.

(2)⇒(3): We want to show that a unique solution to $A\vec{x} = \vec{b}$ implies the matrix A has a trivial null space $\mathcal{N}(A) = \{\vec{0}\}$. We start by assuming the opposite is true: that $\mathcal{N}(A)$ contains at least one nonzero vector $\vec{y} \neq 0$. If this were true, then $\vec{x}' \equiv \vec{x} + \vec{y}$ would also be a solution to $A\vec{x} = \vec{b}$, since $A\vec{x}' = A(\vec{x} + \vec{y}) = A\vec{x} + A\vec{y} = A\vec{x} + \vec{0} = \vec{b}$, since $A\vec{y} = \vec{0}$. The fact that two solutions ($\vec{x}$ and $\vec{x}' \neq \vec{x}$) exist contradicts the statement that $A\vec{x} = \vec{b}$ has a unique solution. Thus, for $A\vec{x} = \vec{b}$ to have a unique solution, A must have a trivial null space $\mathcal{N}(A) = \{\vec{0}\}$.

(3)⇒(4) Statements (**3**) and (**4**) are equivalent by definition: the condition that A's null space contains only the zero vector, $\mathcal{N}(A) = \{\vec{0}\}$, is equivalent to the condition that the only solution to the equation $A\vec{v} = \vec{0}$ is $\vec{v} = \vec{0}$.

(4)⇒(5): Analyze the equation $A\vec{v} = \vec{0}$ in the column picture of matrix multiplication, denoting the n columns of A as $\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_n$. We obtain $A\vec{v} = v_1\vec{c}_1 + v_2\vec{c}_2 + \cdots + v_n\vec{c}_n = \vec{0}$. Since $\vec{v} = (v_1, v_2, \ldots, v_n) = \vec{0}$ is the only solution to this equation, we obtain the statement in the definition of linear independence for a set of vectors. The fact that $A\vec{v} = \vec{0}$ has only $\vec{v} = \vec{0}$ as a solution implies the columns of A form a linearly independent set. Furthermore, the columns of A form a basis for $\mathbb{R}^n$ because they are a set of n linearly independent vectors in an n-dimensional vector space.

(5)⇒(6): We know $\text{rank}(A)$ equals the number of linearly independent columns in A. Since the n columns of A are linearly independent, it follows that $\text{rank}(A) = n$.

(6)⇒(7): The rank is also equal to the number of leading ones (pivots) in the RREF of A. Since A has rank n, its reduced row echelon form must contain n pivots. The reduced row echelon form of an $n \times n$ matrix with n pivots is the identity matrix $\mathbb{1}_n$.

(7)⇒(1): We start from the assumption $\text{rref}(A) = \mathbb{1}_n$. This means it's possible to use a set of row operations $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$ to transform A to the identity matrix: $\mathcal{R}_k(\cdots \mathcal{R}_2(\mathcal{R}_1(A)) \cdots) = \mathbb{1}_n$. Consider the elementary matrices $E_1, E_2, \ldots, E_k$ that correspond to the row operations $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$. Rewriting the equation $\mathcal{R}_k(\cdots \mathcal{R}_2(\mathcal{R}_1(A)) \cdots) = \mathbb{1}_n$ in terms of these elementary matrices gives us $E_k \cdots E_2 E_1 A = \mathbb{1}_n$. This equation implies the inverse of A exists and is equal to the product of elementary matrices $A^{-1} \equiv E_k \cdots E_2 E_1$.

(1)⇔(8): If A is invertible, there exists A^{-1} such that $AA^{-1} = \mathbb{1}_n$. If we apply the transpose operation to this equation, we obtain

$$(AA^{-1})^\mathsf{T} = (\mathbb{1}_n)^\mathsf{T} \qquad \Rightarrow \qquad (A^{-1})^\mathsf{T} A^\mathsf{T} = \mathbb{1}_n,$$

which shows the matrix $(A^{-1})^\mathsf{T} = (A^\mathsf{T})^{-1}$ exists and is the inverse of A^T. Therefore, A is invertible if and only if A^T is invertible.

(8)⇔(9): Statement (**9**) follows by a combination of statements (**8**) and (**5**): if A^T is invertible, its columns form a basis for $\mathbb{R}^n$. Since the columns of A^T are the rows of A, it follows that the rows of A form a basis for $\mathbb{R}^n$.

(5)⇔(10): The determinant of an $n \times n$ matrix is nonzero if and only if the columns of the matrix are linearly independent. Thus, the columns of the matrix A form a basis for $\mathbb{R}^n$ if and only if $\det(A) \neq 0$.

Nice work! By proving the chain of implications (**1**) ⇒ (**2**) ⇒ $\cdots$ ⇒ (**7**) ⇒ (**1**), we've shown that the first seven statements are equivalent. If one of these statements is true, then all others are true—just follow the arrows of implication. Alternatively, if one statement is false, all statements are false, as we see by following the arrows of implication in the backward direction.

We also "attached" statements (**8**), (**9**), and (**10**) to the main loop of implications using "if an only if" statements. Thus, we've shown the equivalence of all 10 statements, which completes the proof. □

The steps of the proof shown above cover only a small selection of all possible implications between the 10 statements. Coming up, you'll be asked to prove several other implications as exercises. It's important to practice these proofs! Obtaining proofs forces your brain to truly grasp about linear algebra concepts and their definitions, as well as directly apply their properties. Note the crucial difference

between "one-way" implications of the form $(\mathbf{A}) \Rightarrow (\mathbf{B})$ and *if and only if* statements $(\mathbf{A}) \Leftrightarrow (\mathbf{B})$. The latter require you to prove both directions of the implication: $(\mathbf{A}) \Rightarrow (\mathbf{B})$ and $(\mathbf{A}) \Leftarrow (\mathbf{B})$.

Invertible linear transformations

We can reinterpret the statements in the **invertible matrix theorem** as a statement about *invertible linear transformations*:

$$T\colon \mathbb{R}^n \to \mathbb{R}^n \text{ is invertible} \quad \Leftrightarrow \quad M_T \in \mathbb{R}^{n\times n} \text{ is invertible.}$$

The set of linear transformations splits into two disjoint subsets: invertible linear transformations and non-invertible linear transformations.

Kernel and null space

The *kernel* of the linear transformation T is the same as the null space of its matrix representation M_T. Recall statement (**3**) of the **invertible matrix theorem**: a matrix A is invertible if and only if its null space contains only the zero vector $\mathcal{N}(A) = \{\vec{0}\}$. The equivalent condition for linear transformations is the zero kernel condition. A linear transformation T is invertible if and only if its kernel contains only the zero vector:

$$\text{Ker}(T) = \{\vec{0}\} \quad \Leftrightarrow \quad T \text{ is invertible.}$$

Invertible linear transformations $T : \mathbb{R}^n \to \mathbb{R}^n$ map different input vectors $\vec{x}$ to different output vectors $\vec{y} \equiv T(\vec{v})$; therefore it's possible to build an inverse linear transformation $T^{-1} : \mathbb{R}^n \to \mathbb{R}^n$ that restores every $\vec{y}$ back to the $\vec{x}$ it came from.

In contrast, a non-invertible linear transformation S sends all vectors $\vec{x} \in \text{Ker}(S)$ to the zero vector $S(\vec{x}) = \vec{0}$. When this happens, there is no way to undo the action of S since we can't determine the original $\vec{x}$ that was sent to $\vec{0}$.

Linear transformations as functions

In Section 1.8, we discussed the notion of *invertibility* for functions of a real variable, $f : \mathbb{R} \to \mathbb{R}$. In particular, we used the terms *injective*, *surjective*, and *bijective* to describe how a function maps different inputs from its domain to outputs in its codomain (see page 33). Since linear transformations are vector functions, we can apply the general terminology for functions to describe how linear transformations map different inputs to outputs.

A linear transformation is *injective* if it maps different inputs to different outputs:

$$T(\vec{v}_1) \neq T(\vec{v}_2) \text{ for all } \vec{v}_1 \neq \vec{v}_2 \quad \Leftrightarrow \quad T \text{ is } \textit{injective}.$$

A linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ is *surjective* if its image space equals its codomain:

$$\text{Im}(T) = \mathbb{R}^m \quad \Leftrightarrow \quad T \text{ is } \textit{surjective}.$$

The surjective condition for linear transformations is equivalent to the condition that the column space of the matrix M_T spans the outputs space: $\text{Im}(T) \equiv \mathcal{C}(M_T) = \mathbb{R}^m$.

If a function is both injective and surjective then it is *bijective*. Bijective functions are *one-to-one correspondences* between elements in their input space and elements in their output space.

$$T(\vec{x}) = \vec{y} \text{ has exactly one solution for each } \vec{y} \quad \Leftrightarrow \quad T \text{ is } \textit{bijective}.$$

All bijective functions are invertible since each output $\vec{y}$ in the output space corresponds to exactly one $\vec{x}$ in the input space.

Interestingly, for a linear transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ to be invertible, the presence of either the injective or surjective property is sufficient. If T is injective, it must have a $\text{Ker}(T) = \{\vec{0}\}$ so it is invertible. If T is surjective, its matrix representation M_T has rank n; the **rank–nullity theorem** $(\text{rank}(M_T) + \text{nullity}(M_T) = n)$ tells us M_T has an empty null space $\mathcal{N}(M_T) \equiv \text{Ker}(T) = \{\vec{0}\}$, making T invertible.

Links

[See Section 2.3 of this page for a proof walkthrough]
`http://math.nyu.edu/~neylon/linalgfall04/project1/jja/group7.htm`

[Nice writeup about the **invertible matrix theorem** with proofs]
`http://bit.ly/InvMatThmProofs`

[Visualization of a two-dimensional linear transformations]
`http://ncase.me/matrix/`

Exercises

E6.3 Prove that statement (**5**) of the **invertible matrix theorem** implies statement (**2**).

E6.4 Prove that statement (**2**) implies statement (**1**).

Discussion

In this chapter, we learned about linear transformations and their matrix representations. The equivalence $T(\vec{x}) \equiv M_T\vec{x}$ is important because it forms a bridge between the abstract notion of a "vector function" and its concrete implementation as a matrix-vector product. Everything you know about matrices can be applied to linear transformations, and everything you know about linear transformations can be applied to matrices. Which is mind-blowing, if you think about it.

We say T is *represented* by the matrix ${}_{B_W}[M_T]_{B_V}$ with respect to the basis B_V for the input space and the basis B_W for the output space. In Section 6.2 we learned about the "probing procedure" for finding matrix representations with respect to the standard basis, while Section 6.3 discussed the notion of change of basis for matrices. Hold tight, because in the next chapter we'll learn about the eigenvalues and eigenvectors of matrices and discuss the *eigendecomposition* of matrices, which is a type of change of basis.

Section 6.4 gave us the **invertible matrix theorem** along with a taste of what it takes to prove formal math statements. It's extra important that you attempt some of the proofs in the exercise section on page 255. Although proofs can be complicated, they're so worth your time because they force you to clarify the definitions and properties of all the math concepts you've encountered thus far. Attempting the proofs in the problems section to find out if you're a linear algebra amateur, or a linear algebra expert.

6.5 Linear transformations problems

P6.1 Find image space $\text{Im}(T)$ for the linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^3$, defined by $T(x, y) = (x, x - y, 2y)$. You are given the standard basis $B_s = \{(1, 0), (0, 1)\}$.

P6.2 Let V be the vector space consisting of all functions of the form $\alpha e^{2x} \cos x + \beta e^{2x} \sin x$. Consider the following linear transformation $L : V \to V$, $L(f) = f' + f$. Find the matrix representing L with respect to the basis $\{e^{2x} \cos x, e^{2x} \sin x\}$.

P6.3 Find the matrix representation of the derivative operator $Dp \equiv \frac{d}{dx}p(x)$. Assume you're working in the vector space of polynomials of degree three $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, represented as coefficient (a_0, a_1, a_2, a_3).

Chapter 7

Theoretical linear algebra

Let's take a trip down memory lane: 150 pages ago, we embarked on a mind-altering journey through the land of linear algebra. We encountered vector and matrix operations. We studied systems of linear equations, solving them with row operations. We covered miles of linear transformations and their matrix representations. With the skills you've acquired to reach this point, you're ready to delve into the abstract, theoretical aspects of linear algebra—that is, since you know all the useful stuff, you can officially move on to the cool stuff. The lessons in this chapter are less concerned with calculations and more about mind expansion.

In math, we often use abstraction to find the commonalities between different mathematical objects. These parallels give us a deeper understanding of the mathematical structures we compare. This chapter extends what we know about the vector space $\mathbb{R}^n$ to the realm of abstract vector spaces of vector-like mathematical objects (Section 7.3). We'll discuss linear independence, find bases, and count dimensions for these abstract vector spaces. Section 7.4 defines abstract inner product operations and uses them to generalize the concept of orthogonality for abstract vectors. Section 7.5 explores the Gram–Schmidt orthogonalization procedure for distilling orthonormal bases from non-orthonormal bases. The final section, Section 7.7, introduces vectors and matrices with complex coefficients. This section also reviews everything we've learned in this book, so be sure to read it even if complex numbers are not required for your course. We'll also work to develop a taxonomy for the different types of matrices according to their properties and applications (Section 7.2). Section 7.6 investigates matrix decompositions—techniques for splitting matrices into products of simpler matrices. We'll start the chapter by discussing the most important decomposition technique: the *eigendecomposition*, which is a way to uncover the "natural basis" of a matrix.

7.1 Eigenvalues and eigenvectors

The set of eigenvectors of a matrix is a special set of input vectors for which the action of the matrix is described as a simple *scaling*. In Section 6.2, we observed how linear transformations act differently on different input spaces. We also observed the special case of the "zero eigenspace," called the *null space* of a matrix. The action of a matrix on the vectors in its null space is equivalent to a multiplication by zero. We'll now put these eigenvalues and eigenvectors under the microscope and see what more there is to see.

Decomposing a matrix in terms of its eigenvalues and its eigenvectors gives valuable insights into the properties of the matrix. Certain matrix calculations, like computing the power of the matrix, become much easier when we use the *eigendecomposition* of the matrix. For example, suppose we're given a square matrix A and want to compute A^7. To make this example more concrete, we'll analyze the matrix

$$A = \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}.$$

We want to compute

$$A^7 = \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}.$$

That's an awful lot of matrix multiplications! Now imagine how many times we'd need to multiply the matrix if we wanted to find A^{17} or A^{77}—*too many* times, that's how many. Let's be smart about this. Every matrix corresponds to some linear operation. This means it's legit to ask, "what does the matrix A do?" Once we figure out this part, we can compute A^{77} by simply doing what A does 77 times.

The best way to see what a matrix does is to look inside it and see what it's made of (you may need to gradually gain the matrix's trust before it lets you do this). To understand the matrix A, you must find its *eigenvectors* and its *eigenvalues* (*eigen* is the German word for "self"). The eigenvectors of a matrix are a "natural basis" for describing the action of the the matrix. The *eigendecomposition* is a change-of-basis operation that expresses the matrix A with respect to its *eigenbasis* (own-basis). The eigendecomposition of the matrix A is a product of three matrices:

$$A = \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \frac{1}{5} & \frac{2}{5} \\ \frac{2}{5} & -\frac{1}{5} \end{bmatrix}}_{Q^{-1}} = Q\Lambda Q^{-1}.$$

You can multiply the three matrices $Q\Lambda Q^{-1}$ to obtain A. Note that the "middle matrix" Λ (the capital Greek letter *lambda*) has entries

only on the diagonal. The diagonal matrix Λ is sandwiched between the matrix Q on the left and Q^{-1} (the inverse of Q) on the right.

The eigendecomposition of A allows us to compute A^7 in a civilized manner:

$$\begin{aligned} A^7 &= AAAAAAA \\ &= Q\Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda \underbrace{Q^{-1}Q}_{\mathbb{1}} \Lambda Q^{-1} \\ &= Q\Lambda\mathbb{1}\Lambda\mathbb{1}\Lambda\mathbb{1}\Lambda\mathbb{1}\Lambda\mathbb{1}\Lambda\mathbb{1}\Lambda Q^{-1} \\ &= Q\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda Q^{-1} \\ &= Q\Lambda^7 Q^{-1}. \end{aligned}$$

All the inner Q^{-1}s cancel with the adjacent Qs. How convenient! Since the matrix Λ is diagonal, it's easy to compute its seventh power:

$$\Lambda^7 = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix}^7 = \begin{bmatrix} 5^7 & 0 \\ 0 & 10^7 \end{bmatrix} = \begin{bmatrix} 78125 & 0 \\ 0 & 10\,000\,000 \end{bmatrix}.$$

Thus we can express our calculation of A^7 as

$$A^7 = \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}^7 = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}}_{Q} \begin{bmatrix} 78125 & 0 \\ 0 & 10\,000\,000 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{5} & \frac{2}{5} \\ \frac{2}{5} & -\frac{1}{5} \end{bmatrix}}_{Q^{-1}}.$$

We still need to multiply these three matrices together, but we've cut the work from six matrix multiplications to two. The answer is

$$A^7 = Q\Lambda^7 Q^{-1} = \begin{bmatrix} 8015625 & -3968750 \\ -3968750 & 2062500 \end{bmatrix}.$$

With this technique, we can compute A^{17} just as easily:

$$A^{17} = Q\Lambda^{17} Q^{-1} = \begin{bmatrix} 80000152587890625 & -39999694824218750 \\ -39999694824218750 & 20000610351562500 \end{bmatrix}.$$

We could even compute $A^{777} = Q\Lambda^{777}Q^{-1}$ if we wanted to. I hope by now you get the point: once you express A in its *eigenbasis*, computed powers of A requires computing powers of its *eigenvalues*, which is much simpler than carrying out matrix multiplications.

Definitions

- A: an $n \times n$ square matrix. The entries of A are denoted as a_{ij}.

- $\text{eig}(A) \equiv (\lambda_1, \lambda_2, \ldots, \lambda_n)$: the list of *eigenvalues* of A. Eigenvalues are usually denoted by the Greek letter *lambda*. Note that some eigenvalues could be repeated in the list.
- $p(\lambda) = \det(A - \lambda\mathbb{1})$: the *characteristic polynomial* of A. The eigenvalues of A are the roots of the characteristic polynomial.
- $\{\vec{e}_{\lambda_1}, \vec{e}_{\lambda_2}, \ldots, \vec{e}_{\lambda_n}\}$: the set of *eigenvectors* of A. Each eigenvector is associated with a corresponding eigenvalue.
- $\Lambda \equiv \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$: the diagonalized version of A. The matrix Λ contains the eigenvalues of A on the diagonal:

$$\Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}.$$

  The matrix Λ is the matrix A expressed in its eigenbasis.
- Q: a matrix whose columns are eigenvectors of A:

$$Q \equiv \begin{bmatrix} | & & | \\ \vec{e}_{\lambda_1} & \cdots & \vec{e}_{\lambda_n} \\ | & & | \end{bmatrix} = {}_{B_s}[\mathbb{1}]_{B_\lambda}.$$

  The matrix Q corresponds to the *change-of-basis* matrix from the eigenbasis $B_\lambda = \{\vec{e}_{\lambda_1}, \vec{e}_{\lambda_2}, \vec{e}_{\lambda_3}, \ldots\}$ to the standard basis $B_s = \{\hat{\imath}, \hat{\jmath}, \hat{k}, \ldots\}$.
- $A = Q\Lambda Q^{-1}$: the *eigendecomposition* of the matrix A
- $\Lambda = Q^{-1}AQ$: the *diagonalization* of the matrix A

Eigenvalues

The fundamental eigenvalue equation is

$$A\vec{e}_\lambda = \lambda\vec{e}_\lambda,$$

where λ is an eigenvalue and $\vec{e}_\lambda$ is an eigenvector of the matrix A. Multiply A by one of its eigenvectors $\vec{e}_\lambda$, and the result is the same vector scaled by the constant λ.

To find the eigenvalues of a matrix, start from the eigenvalue equation $A\vec{e}_\lambda = \lambda\vec{e}_\lambda$, insert the identity $\mathbb{1}$, and rewrite the equation as a null-space problem:

$$A\vec{e}_\lambda = \lambda\mathbb{1}\vec{e}_\lambda \qquad \Rightarrow \qquad (A - \lambda\mathbb{1})\,\vec{e}_\lambda = \vec{0}.$$

This equation has a solution whenever $|A - \lambda\mathbb{1}| = 0$. The eigenvalues of $A \in \mathbb{R}^{n\times n}$, denoted $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, are the roots of the *characteristic polynomial*:

$$p(\lambda) \equiv \det(A - \lambda\mathbb{1}) = 0.$$

Calculate this determinant and we obtain an expression involving the coefficients a_{ij} and the variable λ. If A is an $n \times n$ matrix, the characteristic polynomial is a polynomial of degree n in λ.

We denote the list of eigenvalues as $\text{eig}(A) = (\lambda_1, \lambda_2, \ldots, \lambda_n)$. If λ_i is a repeated root of the characteristic polynomial $p(\lambda)$, it's called a *degenerate* eigenvalue. For example, the identity matrix $\mathbb{1} \in \mathbb{R}^{2\times 2}$ has the characteristic polynomial $p_{\mathbb{1}}(\lambda) = (\lambda - 1)^2$, which has a repeated root at $\lambda = 1$. We say the eigenvalue $\lambda = 1$ is *degenerate* and has *algebraic multiplicity* 2. It's important to keep track of degenerate eigenvalues, so we specify the multiplicity of an eigenvalue by repeatedly including it in the list of eigenvalues: $\text{eig}(\mathbb{1}) = (\lambda_1, \lambda_2) = (1, 1)$.

Eigenvectors

The *eigenvectors* associated with eigenvalue λ_i of matrix A are the vectors in the *null space* of the matrix $(A - \lambda_i \mathbb{1})$.

To find the eigenvectors associated with the eigenvalue λ_i, you need to solve for the components $e_{\lambda,x}$ and $e_{\lambda,y}$ of the vector $\vec{e}_\lambda = (e_{\lambda,x}, e_{\lambda,y})$ that satisfies the equation

$$A\vec{e}_\lambda = \lambda\vec{e}_\lambda,$$

or equivalently,

$$(A - \lambda\mathbb{1})\vec{e}_\lambda = 0 \qquad \Leftrightarrow \qquad \begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} \begin{bmatrix} e_{\lambda,x} \\ e_{\lambda,y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

You previously solved this type of problem when you learned how to compute the null space of a matrix.

If λ_i is a repeated root (*degenerate* eigenvalue), the null space of $(A - \lambda_i \mathbb{1})$ could contain multiple eigenvectors. The dimension of the null space of $(A - \lambda_i \mathbb{1})$ is called the *geometric multiplicity* of the eigenvalue λ_i.

Eigendecomposition

If an $n \times n$ matrix A is *diagonalizable* (no, I did't make that word up) this means we can find n eigenvectors for that matrix. The eigenvectors that come from different eigenspaces are guaranteed to be linearly independent (see exercise **E7.3**). We can also pick a set of linearly independent vectors *within* each of the degenerate eigenspaces. Combining the eigenvectors from all the eigenspaces gives us a set of n linearly independent eigenvectors, which form a *basis* for $\mathbb{R}^n$. This is the *eigenbasis*.

Let's place the n eigenvectors side by side as the columns of a matrix:

$$Q \equiv \begin{bmatrix} | & & | \\ \vec{e}_{\lambda_1} & \cdots & \vec{e}_{\lambda_n} \\ | & & | \end{bmatrix}.$$

We can decompose A in terms of its eigenvalues and its eigenvectors:

$$A = Q\Lambda Q^{-1} = \begin{bmatrix} | & & | \\ \vec{e}_{\lambda_1} & \cdots & \vec{e}_{\lambda_n} \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} & Q^{-1} & \end{bmatrix}.$$

The matrix Λ is a diagonal matrix of eigenvalues, and the matrix Q is the change-of-basis matrix that contains the corresponding eigenvectors as columns.

Note that only the *direction* of each eigenvector is important and not the length. Indeed, if $\vec{e}_\lambda$ is an eigenvector (with eigenvalue λ), so is any $\alpha\vec{e}_\lambda$ for all $\alpha \in \mathbb{R}$. Thus we're free to use any multiple of the vectors $\vec{e}_{\lambda_i}$ as the columns of the matrix Q.

Example Find the eigendecomposition of the matrix:

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix}.$$

In decreasing order, the eigenvalues of the matrix are:

$$\lambda_1 = 3, \qquad \lambda_2 = 2, \qquad \lambda_3 = 1.$$

The eigenvalues of A are the values that appear on the diagonal of Λ.

When a 3×3 matrix has three distinct eigenvalues, it is diagonalizable, since it has the same number of linearly independent eigenvectors as eigenvalues. We know the eigenvectors are linearly independent by the following reasoning. The matrix A has three different eigenvalues. Each eigenvalue is associated with at least one eigenvector, and these eigenvectors are linearly independent (see **E7.3** on page 270). Recall that any set of n linearly independent vectors in $\mathbb{R}^n$ forms a basis for $\mathbb{R}^n$. Since the three eigenvectors of A are linearly independent, we have enough columns to construct a change-of-basis matrix of eigenvectors Q and use it to write $A = Q\Lambda Q^{-1}$.

To find the eigenvectors of A, solve for the null space of the matrices $(A - 3\mathbb{1})$, $(A - 2\mathbb{1})$, and $(A - \mathbb{1})$ respectively:

$$\vec{e}_{\lambda_1} = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}, \qquad \vec{e}_{\lambda_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \vec{e}_{\lambda_3} = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}.$$

Check that $A\vec{e}_{\lambda_k} = \lambda_k \vec{e}_{\lambda_k}$ for each of the vectors above. Let Q be the matrix constructed with the eigenvectors as its columns:

$$Q = \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix}, \qquad \text{then compute} \qquad Q^{-1} = \begin{bmatrix} 0 & -1 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}.$$

Together with the matrix Λ, these matrices form the eigendecomposition of the matrix A:

$$A = Q\Lambda Q^{-1} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix} = \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}.$$

To find the diagonalization of A, move Q and Q^{-1} to the other side of the equation. More specifically, multiply the equation $A = Q\Lambda Q^{-1}$ by Q^{-1} on the left and by Q on the right to obtain the diagonal matrix:

$$\Lambda = Q^{-1}AQ = \begin{bmatrix} 0 & -1 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Have you noticed how time consuming it is to compute the eigendecomposition of a matrix? It's really incredible. Though we skipped some details of the calculation (including finding the solution of the characteristic polynomial and the three null space calculations), finding the eigendecomposition of a 3×3 matrix took more than a page of work. Don't be surprised if it takes hours to compute eigendecompositions on homework problems; you're not doing anything wrong. Eigendecompositions simply take a lot of work. We're dealing with a seven-syllable word here, folks—did you really expect the process to be easy?

Well, actually, computing eigenvectors and eigenvalues will become easier with practice. After eigendecomposing a dozen matrices or so using only pen and paper, you'll be able to whip through the steps for a 3×3 matrix in about 20 minutes. That's a really good thing for your GPA, because the probability of seeing an eigenvalue question on your linear algebra final exam is about 80%. It pays to have a mathematical edge with this stuff.

For readers learning linear algebra without the added motivational bonus of exam stress, I recommend you eigendecompose at least one matrix using only pen and paper to prove to yourself you can do it. In all other circumstances, you're better off using `SymPy` to create a `Matrix` object; then calling its `eigenvals()` method to find the eigenvalues, or its `eigenvects()` method to find both its eigenvalues and eigenvectors.

Explanations

Yes, we've got some explaining to do.

Eigenspaces

Recall the definition of the *null space* of a matrix A:

$$\mathcal{N}(A) \equiv \{\vec{v} \in \mathbb{R}^n \mid A\vec{v} = 0\}.$$

The *dimension* of the null space is the number of linearly independent vectors in the null space.

Example If the matrix A sends exactly two linearly independent vectors $\vec{v}$ and $\vec{w}$ to the zero vector—$A\vec{v} = 0,\ A\vec{w} = 0$—then its null space is two-dimensional. We can always choose the vectors $\vec{v}$ and $\vec{w}$ to be orthogonal $\vec{v} \cdot \vec{w} = 0$ and thus obtain an *orthogonal basis* for the null space.

Each eigenvalue λ_i is associated with an *eigenspace*. The eigenspace E_{λ_i} is the null space of the matrix $(A - \lambda_i \mathbb{1})$:

$$E_{\lambda_i} \equiv \mathcal{N}\left(A - \lambda_i \mathbb{1}\right) = \{\vec{v} \in \mathbb{R}^n \mid \left(A - \lambda_i \mathbb{1}\right)\vec{v} = \vec{0}\}.$$

Every eigenspace contains at least one nonzero eigenvector. For degenerate eigenvalues (repeated roots of the characteristic polynomial) the null space of $(A - \lambda_i \mathbb{1})$ can contain multiple eigenvectors.

Change-of-basis matrix

The matrix Q is a *change-of-basis* matrix. Given a vector expressed in the eigenbasis $[\vec{v}]_{B_\lambda} = (v_1', v_2', v_3')^{\mathsf{T}}_{B_\lambda} = v_1'\vec{e}_{\lambda_1} + v_2'\vec{e}_{\lambda_3} + v_3'\vec{e}_{\lambda_3}$, we can use the matrix Q to convert it to coefficients in the standard basis $[\vec{v}]_{B_s} = (v_1, v_2, v_3)^{\mathsf{T}}_{B_s} = v_1\hat{\imath} + v_2\hat{\jmath} + v_3\hat{k}$ as follows:

$$[\vec{v}]_{B_s} = Q[\vec{v}]_{B_\lambda} = {}_{B_s}[\mathbb{1}]_{B_\lambda}[\vec{v}]_{B_\lambda}.$$

The change of basis in the other direction is given by the inverse matrix:

$$[\vec{v}]_{B_\lambda} = \ Q^{-1}[\vec{v}]_{B_s} = {}_{B_\lambda}[\mathbb{1}]_{B_s}[\vec{v}]_{B_s}.$$

Interpretation

The eigendecomposition $A = Q\Lambda Q^{-1}$ allows us to interpret the action of A on an arbitrary input vector $\vec{v}$ as the following three steps:

$$[\vec{w}]_{B_s} = {}_{B_s}[A]_{B_s}[\vec{v}]_{B_s} = Q\Lambda Q^{-1}[\vec{v}]_{B_s} = \underbrace{{}_{B_s}[\mathbb{1}]_{B_\lambda}}_{3}\ \underbrace{{}_{B_\lambda}[\Lambda]_{B_\lambda}}_{2}\ \underbrace{\underbrace{{}_{B_\lambda}[\mathbb{1}]_{B_s}}_{1}[\vec{v}]_{B_s}}_{2}\underbrace{\phantom{{}_{B_\lambda}[\mathbb{1}]_{B_s}}}_{3}.$$

1. In the first step, we convert the vector $\vec{v}$ from the standard basis to the eigenbasis of A.

2. In the second step, the action of A on vectors expressed with respect to its eigenbasis corresponds to a multiplication by the diagonal matrix Λ.

3. In the third step, we convert the output $\vec{w}$ from the eigenbasis back to the standard basis.

Another way to interpret these three steps is to say that, deep down inside, the matrix A is actually the diagonal matrix Λ. To see the diagonal form of the matrix, we must express the input vectors with respect to the *eigenbasis*:

$$[\vec{w}]_{B_\lambda} = {}_{B_\lambda}[\Lambda]_{B_\lambda}[\vec{v}]_{B_\lambda}.$$

It's extremely important you understand the meaning of the equation $A = Q\Lambda Q^{-1}$ intuitively in terms of the three-step procedure. To help understand the three-step procedure, we'll analyze in detail what happens when we multiply A by one of its eigenvectors. Let's pick $\vec{e}_{\lambda_1}$ and verify the equation $A\vec{e}_{\lambda_1} = Q\Lambda Q^{-1}\vec{e}_{\lambda_1} = \lambda_1\vec{e}_{\lambda_1}$ by following the vector through the three steps:

$$\begin{aligned}
{}_{B_s}[A]_{B_s}[\vec{e}_{\lambda_1}]_{B_s} &= Q\Lambda Q^{-1}[\vec{e}_{\lambda_1}]_{B_s} \\
&= {}_{B_s}[\mathbb{1}]_{B_\lambda}\ \underbrace{{}_{B_\lambda}[\Lambda]_{B_\lambda}\ \underbrace{{}_{B_\lambda}[\mathbb{1}]_{B_s}[\vec{e}_{\lambda_1}]_{B_s}}_{(1,0,\ldots)^\mathsf{T}_{B_\lambda}}}_{(\lambda_1,0,\ldots)^\mathsf{T}_{B_\lambda}} = \lambda_1[\vec{e}_{\lambda_1}]_{B_s}.
\end{aligned}$$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\lambda_1[\vec{e}_{\lambda_1}]_{B_s}}$

In the first step, we convert the vector $[\vec{e}_{\lambda_1}]_{B_s}$ to the eigenbasis and obtain $(1, 0, \ldots, 0)^\mathsf{T}_{B_\lambda}$. The second step results in $(\lambda_1, 0, \ldots, 0)^\mathsf{T}_{B_\lambda}$, because multiplying Λ by the vector $(1, 0, \ldots, 0)^\mathsf{T}_{B_\lambda}$ selects the value in the first column of Λ. For the third step, we convert $(\lambda_1, 0, \ldots, 0)^\mathsf{T}_{B_\lambda} = \lambda_1(1, 0, \ldots, 0)^\mathsf{T}_{B_\lambda}$ back to the standard basis to obtain $\lambda_1[\vec{e}_{\lambda_1}]_{B_s}$. Boom!

Invariant properties of matrices

The determinant and the trace of a matrix are strictly functions of the eigenvalues. The determinant of A is the product of its eigenvalues:

$$\det(A) \equiv |A| = \prod_i \lambda_i = \lambda_1\lambda_2\cdots\lambda_n,$$

and the trace is the sum of the eigenvalues:

$$\text{Tr}(A) = \sum_i a_{ii} = \sum_i \lambda_i = \lambda_1 + \lambda_2 + \cdots \lambda_n.$$

The above equations are true because

$$|A| = |Q\Lambda Q^{-1}| = |Q||\Lambda||Q^{-1}| = |Q||Q^{-1}||\Lambda| = \frac{|Q|}{|Q|}|\Lambda| = |\Lambda| = \prod_i \lambda_i,$$

and

$$\text{Tr}(A) = \text{Tr}(Q\Lambda Q^{-1}) = \text{Tr}(\Lambda Q^{-1}Q) = \text{Tr}(\Lambda \mathbb{1}) = \text{Tr}(\Lambda) = \sum_i \lambda_i.$$

The first equation follows from the properties of determinants: $|AB| = |A||B|$ and $|A^{-1}| = \frac{1}{|A|}$ (see page 125). The second equation follows from the cyclic property of the trace operator $\text{Tr}(ABC) = \text{Tr}(BCA)$ (see page 125).

In fact, the above calculations are true for any *similarity transformation*. Recall that a similarity transformation is a change-of-basis calculation in which a matrix A gets multiplied by an invertible matrix P from the left and by the inverse P^{-1} from the right: $A' = PAP^{-1}$. The determinant and the trace of a matrix are *invariant* properties under similarity transformations—they don't depend on the choice of basis.

Relation to invertibility

Let's briefly revisit three of the equivalent conditions we stated in the **invertible matrix theorem**. For a matrix $A \in \mathbb{R}^{n\times n}$, the following statements are equivalent:

- A is invertible
- $|A| \neq 0$
- The null space contains only the zero vector $\mathcal{N}(A) = \{\vec{0}\}$

The formula $|A| = \lambda_1\lambda_2\cdots\lambda_n$ reveals why the last two statements are equivalent. If $|A| \neq 0$, none of the λ_is are zero (if one of the eigenvalues is zero, the whole product is zero). We know $\lambda = 0$ is *not* an eigenvalue of A, which means there exists no vector $\vec{v}$ such that $A\vec{v} = 0\vec{v} = \vec{0}$. Therefore, there are no vectors in the null space $\mathcal{N}(A)$. We can also follow this reasoning in the other direction. If the null space of A is empty, then there is no nonzero vector $\vec{v}$ such that $A\vec{v} = 0\vec{v} = \vec{0}$, which means $\lambda = 0$ is not an eigenvalue of A, hence the product $\lambda_1\lambda_2\cdots\lambda_n \neq 0$.

However, if there exists a nonzero vector $\vec{v}$ such that $A\vec{v} = \vec{0}$, then A has a non-empty null space, $\lambda = 0$ is an eigenvalue of A, and thus $|A| = \lambda_1\lambda_2\cdots\lambda_n = 0$.

Eigendecomposition for normal matrices

A matrix A is *normal* if it satisfies the equation $A^\mathsf{T}A = AA^\mathsf{T}$. All normal matrices are diagonalizable, and the change-of-basis matrix Q can be chosen to be an *orthogonal* matrix O.

The eigenvectors corresponding to different eigenvalues of a normal matrix are *orthogonal*. Furthermore, we can choose the eigenvectors within the same eigenspace to be orthogonal. By collecting the eigenvectors from all eigenspaces of the matrix $A \in \mathbb{R}^{n\times n}$, it is possible to obtain a basis $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ of orthogonal eigenvectors:

$$\vec{e}_i \cdot \vec{e}_j = \begin{cases} \|\vec{e}_i\|^2 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Normalizing these vectors gives a set of *orthonormal* eigenvectors $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ that form a basis for the space $\mathbb{R}^n$:

$$\hat{e}_i \cdot \hat{e}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Consider now the matrix O constructed by using these orthonormal vectors as the columns:

$$O = \begin{bmatrix} | & & | \\ \hat{e}_1 & \cdots & \hat{e}_n \\ | & & | \end{bmatrix}.$$

The matrix O is an *orthogonal* matrix, meaning it satisfies $OO^\mathsf{T} = I = O^\mathsf{T}O$. In other words, the inverse of O is obtained by taking the transpose O^T. To see how this works, consider the following product:

$$O^\mathsf{T}O = \begin{bmatrix} — & \hat{e}_1 & — \\ & \vdots & \\ — & \hat{e}_n & — \end{bmatrix} \begin{bmatrix} | & & | \\ \hat{e}_1 & \cdots & \hat{e}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbb{1}.$$

Each of the ones on the diagonal arises from taking the dot product of a unit-length eigenvector with itself. The off-diagonal entries are zero because the vectors are orthogonal. By definition, the inverse O^{-1} is the matrix, which gives $\mathbb{1}$ when multiplied by O, so we have $O^{-1} = O^\mathsf{T}$.

Using the orthogonal matrix O and its inverse O^T, we can write the eigendecomposition of a matrix A as

$$A = O\Lambda O^{-1} = O\Lambda O^\mathsf{T} = \begin{bmatrix} | & & | \\ \hat{e}_1 & \cdots & \hat{e}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} — & \hat{e}_1 & — \\ & \vdots & \\ — & \hat{e}_n & — \end{bmatrix}.$$

The key advantage of using an orthogonal matrix O in the diagonalization procedure is that computing its inverse becomes a trivial task: $O^{-1} = O^{\mathsf{T}}$. The class of normal matrices enjoy a special status by virtue of being diagonalizable by orthogonal matrices.

Discussion

Non-diagonalizable matrices

Not all matrices are diagonalizable. For example, the matrix

$$B = \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix}$$

has $\lambda = 3$ as a repeated eigenvalue, but the null space of the matrix $(B - 3\mathbb{1})$ contains only one eigenvector: $(1, 0)^{\mathsf{T}}$. The matrix B has a single eigenvector in the eigenspace $\lambda = 3$. To describe this situation using precise mathematical terminology, we say the *algebraic multiplicity* of the eigenvalue $\lambda = 3$ is two, but the *geometric multiplicity* of the eigenvalue is one.

The matrix B is a 2×2 matrix with a single eigenvector. Since we're one eigenvector short, we can't construct the diagonalizing change-of-basis matrix Q. We say the matrix has *deficient* geometric multiplicity, meaning it doesn't have a full set of eigenvectors. Therefore, B is not diagonalizable.

Matrix power series

One of the most useful concepts of calculus is the idea that functions can be represented as Taylor series. The Taylor series of the exponential function $f(x) = e^x$ is

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots .$$

Nothing stops us from using the same Taylor series expression to define the exponential function of a matrix:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = 1 + A + \frac{A^2}{2} + \frac{A^3}{3!} + \frac{A^4}{4!} + \frac{A^5}{5!} + \cdots .$$

Okay, there *is* one thing stopping us—we need to compute an infinite sum of progressively larger matrix powers. Remember how we used the diagonalization of $A = Q\Lambda Q^{-1}$ to write A^{77} as $Q\Lambda^{77}Q^{-1}$? We can apply that trick here to obtain the exponential of a matrix in a much simpler form:

$$
\begin{aligned}
e^A &= \sum_{k=0}^{\infty} \frac{A^k}{k!} = \sum_{k=0}^{\infty} \frac{(Q\Lambda Q^{-1})^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{Q\,\Lambda^k\,Q^{-1}}{k!} \\
&= Q\left[\sum_{k=0}^{\infty} \frac{\Lambda^k}{k!}\right] Q^{-1} \\
&= Q\left(1 + \Lambda + \frac{\Lambda^2}{2} + \frac{\Lambda^3}{3!} + \frac{\Lambda^4}{4!} + \ldots\right) Q^{-1} \\
&= Qe^{\Lambda}Q^{-1} \\
&= \begin{bmatrix} & & \\ & Q & \\ & & \end{bmatrix} \begin{bmatrix} e^{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & e^{\lambda_n} \end{bmatrix} \begin{bmatrix} & & \\ & Q^{-1} & \\ & & \end{bmatrix}.
\end{aligned}
$$

We can use this approach to define "matrix functions" of the form

$$F : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n\times n}$$

as **Taylor series of matrices**. Computing the matrix function $F(A)$ on an input matrix $A = Q\Lambda Q^{-1}$ is equivalent to computing the function f of each of the eigenvalues of the matrix: $F(A) = Q\,f(\Lambda)\,Q^{-1}$.

Review

We learned how to decompose matrices in terms of their eigenvalues and eigenvectors. The fundamental equation is $A\vec{e}_\lambda = \lambda\vec{e}_\lambda$, where the vector $\vec{e}_\lambda$ is an *eigenvector* of the matrix A, and the number λ is an *eigenvalue* of A. The word *eigen* is the German word for "self."

The characteristic polynomial comes about from a simple manipulation of the eigenvalue equation:

$$
\begin{aligned}
A\vec{e}_\lambda &= \lambda\vec{e}_\lambda \\
A\vec{e}_\lambda - \lambda\vec{e}_\lambda &= 0 \\
(A - \lambda\mathbb{1})\vec{e}_\lambda &= 0.
\end{aligned}
$$

For this equation to be satisfied, the vector $\vec{e}_\lambda$ must be in the *null space* of $(A - \lambda\mathbb{1})$. The problem of finding the eigenvalues reduces to finding the values of λ for which the matrix $(A - \lambda\mathbb{1})$ has a non-empty null space. Recall that a matrix has a non-empty null space if and only if it is not invertible. The easiest way to check if a matrix is invertible is to compute the determinant: $|A - \lambda\mathbb{1}| = 0$.

Because multiple eigenvalues and eigenvectors may satisfy this equation, we keep a list of eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and corresponding eigenvectors $\{\vec{e}_{\lambda_1}, \vec{e}_{\lambda_2}, \ldots\}$. The eigendecomposition of the matrix is $A = Q\Lambda Q^{-1}$, where Q is the matrix with eigenvectors as columns and Λ contains the eigenvalues on the diagonal.

Applications

Many scientific methods use the eigendecomposition of a matrix as a building block. For instance:

- In statistics, the *principal component analysis* technique aims to uncover the dominant cause of the variation in datasets by eigendecomposing the *covariance matrix*—a matrix computed from the dataset.
- Google's original `PageRank` algorithm for ranking webpages by "importance" can be explained as the search for an eigenvector of a matrix. The matrix contains information about all the hyperlinks that exist between webpages (see Section 9.3).
- In quantum mechanics, the energy of a system is described by the Hamiltonian operator. The eigenvalues of the Hamiltonian are the possible energy levels the system can have.

Analyzing a matrix in terms of its eigenvalues and its eigenvectors is a powerful technique to "see inside" a matrix and understand what the matrix does. In the next section, we'll analyze several different types of matrices and discuss their properties in terms of their eigenvalues.

Links

[Good visual examples of eigenvectors from Wikipedia]
`http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors`

Exercises

E7.1 Explain why an $n \times n$ matrix A can have at most n different eigenvalues.

E7.2 Check out Problems 1 through 5 at the following URL:
`en.wikibooks.org/wiki/Linear_Algebra/Eigenvalues_and_Eigenvectors`
`en.wikibooks.org/wiki/Linear_Algebra/Eigenvalues_and_Eigenvectors/Solutions`

E7.3 Prove that the eigenvectors that correspond to different eigenvalues are linearly independent.

E7.4 Let λ be an eigenvalue of A and let $\vec{e}_\lambda$ be the corresponding eigenvector. Show that λ^2 is an eigenvalue of A^2.

E7.5 Suppose λ is an eigenvalue of the invertible matrix A with corresponding eigenvector $\vec{e}_\lambda$. Show that λ^{-1} is an eigenvalue of the inverse matrix A^{-1}.

E7.6 Find the values of α and β so the matrix $A = \begin{bmatrix} 0 & \alpha \\ 1 & \beta \end{bmatrix}$ will have eigenvalues 1 and 3.

E7.7 Consider the matrix $L = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$. Which of the following vectors are eigenvectors of L?

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ -1 \end{bmatrix}.$$

7.2 Special types of matrices

Mathematicians just love to categorize things. Conveniently for us, they've categorized certain types of matrices. Rather than embarking on a verbose explanation of the properties of a matrix, such as

```
I have this matrix A whose rows are perpendicular vectors and
when you multiply any vector by this matrix it doesn't change
the length of the vector but just kind of rotates it---
```

it's much simpler to refer to the categorization by saying,

```
Let A be an orthogonal matrix.
```

Most advanced science textbooks and research papers routinely use terminology like "diagonal matrix," "symmetric matrix," and "orthogonal matrix," so make sure you're familiar with these concepts.

This section will also review and reinforce what we learned about linear transformations. Recall that we can think of the matrix-vector product $A\vec{x}$ as applying a linear transformation T_A to an input vector $\vec{x}$. Therefore, each of the special matrices discussed here also corresponds to a special type of linear transformation. Keep this dual correspondence in mind because we'll use the same terminology to describe matrices *and* linear transformations.

Notation

- $\mathbb{R}^{m\times n}$: the set of $m \times n$ matrices
- $A, B, C, O, P, Q, \ldots$: typical names for matrices
- a_{ij}: the entry in the i^{th} row and j^{th} column of the matrix A
- A^{T}: the transpose of the matrix A

- A^{-1}: the inverse of the matrix A
- $\lambda_1, \lambda_2, \ldots$: the *eigenvalues* of the matrix A. For each eigenvalue λ_i there is at least one associated *eigenvector* $\vec{e}_{\lambda_i}$ that obeys the equation $A\vec{e}_{\lambda_i} = \lambda_i \vec{e}_{\lambda_i}$. Multiplying the matrix A by its eigenvectors $\vec{e}_{\lambda_i}$ is the same as scaling $\vec{e}_{\lambda_i}$ by λ_i.

Diagonal matrices

Diagonal matrices contain entries on the diagonal and zeros everywhere else. For example:

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}.$$

A diagonal matrix A satisfies $a_{ij} = 0, \text{if } i \neq j$. The eigenvalues of a diagonal matrix are $\lambda_i = a_{ii}$.

Symmetric matrices

A matrix A is symmetric if and only if

$$A^{\mathsf{T}} = A, \quad \text{or equivalently if} \quad a_{ij} = a_{ji}, \text{ for all } i, j.$$

The eigenvalues of symmetric matrices are real numbers, and the eigenvectors can be chosen to be mutually orthogonal.

Given any matrix $B \in \mathbb{R}^{m \times n}$, the product of B with its transpose $B^{\mathsf{T}}B$ is always a symmetric matrix.

Upper triangular matrices

Upper triangular matrices have zero entries below the main diagonal:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}, \qquad a_{ij} = 0, \quad \text{if } i > j.$$

For a *lower* triangular matrix, all the entries *above* the diagonal are zeros: $a_{ij} = 0, \text{if } i < j$.

Identity matrix

The identity matrix is denoted $\mathbb{1}$ or $\mathbb{1}_n \in \mathbb{R}^{n \times n}$ and plays the role of multiplication by the number 1 for matrices: $\mathbb{1}A = A\mathbb{1} = A$. The identity matrix is diagonal with ones on the diagonal:

$$\mathbb{1}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Any vector $\vec{v} \in \mathbb{R}^3$ is an eigenvector of the identity matrix with eigenvalue $\lambda = 1$.

Orthogonal matrices

A matrix $O \in \mathbb{R}^{n \times n}$ is *orthogonal* if it satisfies $OO^\mathsf{T} = \mathbb{1} = O^\mathsf{T}O$. In other words, the inverse of an orthogonal matrix O is obtained by taking its transpose: $O^{-1} = O^\mathsf{T}$.

Multiplication by an orthogonal matrix preserves lengths. Consider the matrix-vector product $O\vec{v} = \vec{w}$. The length of a vector before the multiplication is $\|\vec{v}\| = \sqrt{\vec{v} \cdot \vec{v}}$. The length of a vector after the multiplication is

$$\|\vec{w}\| = \sqrt{\vec{w} \cdot \vec{w}} = \sqrt{(O\vec{v})^\mathsf{T}(O\vec{v})} = \sqrt{\vec{v}^\mathsf{T}O^\mathsf{T}O\vec{v}}.$$

The second equality follows from the interpretation of the dot product as a matrix product $\vec{u} \cdot \vec{v} = \vec{u}^\mathsf{T}\vec{v}$. The third equality follows from the properties of matrix transpose $(AB)^\mathsf{T} = B^\mathsf{T}A^\mathsf{T}$.

When O is an orthogonal matrix, we can substitute $O^\mathsf{T}O = \mathbb{1}$ in the above expression to establish $\|\vec{w}\| = \sqrt{\vec{v}^\mathsf{T}\mathbb{1}\vec{v}} = \|\vec{v}\|$, which shows that multiplication by an orthogonal matrix is a *length preserving* operation.

The eigenvalues of an orthogonal matrix have *unit* length, but can in general be complex numbers $\lambda = e^{i\theta} \in \mathbb{C}$. The determinant of an orthogonal matrix is either one or negative one $|O| \in \{-1, 1\}$.

You can visualize orthogonal matrices by thinking of their columns as a set of vectors that form an orthonormal basis for $\mathbb{R}^n$:

$$O = \begin{bmatrix} | & & | \\ \hat{e}_1 & \cdots & \hat{e}_n \\ | & & | \end{bmatrix} \quad \text{such that} \quad \hat{e}_i \cdot \hat{e}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

You can verify the matrix O is orthogonal by computing $O^\mathsf{T}O = \mathbb{1}$. The orthogonal matrix O is a change-of-basis matrix from the standard basis to the "column basis" $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$.

Everything stated above about multiplication by an orthogonal matrix also applies to orthogonal transformations $T_O : \mathbb{R}^n \to \mathbb{R}^n$ because of the equivalence $O\vec{v} = \vec{w} \;\Leftrightarrow\; T_O(\vec{v}) = \vec{w}$.

The set of orthogonal matrices contains three special cases: *rotations* matrices, *reflection* matrices, and *permutation* matrices.

Rotation matrices

A rotation matrix takes the standard basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ to a rotated basis $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$. Consider an example in $\mathbb{R}^2$. The counterclockwise

rotation by the angle θ is given by the matrix

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

The matrix R_θ takes $\hat{\imath} = (1,0)^\mathsf{T}$ to $(\cos\theta, \sin\theta)^\mathsf{T}$, and $\hat{\jmath} = (0,1)^\mathsf{T}$ to $(-\sin\theta, \cos\theta)^\mathsf{T}$.

As another example, consider the rotation by the angle θ around the x-axis in $\mathbb{R}^3$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}.$$

This rotation is entirely in the yz-plane, so the x-component of a vector multiplying this matrix remains unchanged.

The determinant of a rotation matrix is equal to one. The eigenvalues of rotation matrices are complex numbers with unit magnitude.

Reflections

If the determinant of an orthogonal matrix O is equal to negative one, we say it is *mirrored orthogonal*. For example, the reflection through the line with direction vector $(\cos\theta, \sin\theta)$ is given by:

$$R = \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix}.$$

A reflection matrix always has at least one eigenvalue equal to negative one, which corresponds to the direction perpendicular to the axis of reflection.

Permutation matrices

Permutation matrices are another important class of orthogonal matrices. The action of a permutation matrix is simply to change the *order* of the coefficients of a vector. For example, the permutation $\pi : \hat{e}_1 \to \hat{e}'_1$, $\hat{e}_2 \to \hat{e}'_3$, $\hat{e}_3 \to \hat{e}'_2$ can be represented as the matrix

$$M_\pi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

An $n \times n$ permutation matrix contains n ones in n different columns and zeros everywhere else.

The *sign* of a permutation corresponds to the determinant $\det(M_\pi)$. We say that permutation π is *even* if $\det(M_\pi) = +1$ and *odd* if $\det(M_\pi) = -1$.

Positive matrices

A matrix $P \in \mathbb{R}^{n \times n}$ is *positive semidefinite* if

$$\vec{v}^{\mathsf{T}} P \vec{v} \geq 0 \quad \text{for all } \vec{v} \in \mathbb{R}^n.$$

The eigenvalues of a positive semidefinite matrix are all nonnegative $\lambda_i \geq 0$.

If instead the matrix P obeys the strict inequality $\vec{v}^{\mathsf{T}} P \vec{v} > 0$ for all $\vec{v} \in \mathbb{R}^n$, we say the matrix P is *positive definite*. The eigenvalues of positive definite matrices are strictly greater than zero $\lambda_i > 0$.

Projection matrices

The defining property of a projection matrix is that it can be applied multiple times without changing the result:

$$\Pi = \Pi^2 = \Pi^3 = \Pi^4 = \Pi^5 = \cdots .$$

A projection has two eigenvalues: one and zero. The space S that is left invariant by the projection Π_S corresponds to the eigenvalue $\lambda = 1$. The orthogonal complement $S^\perp$ corresponds to the eigenvalue $\lambda = 0$ and consists of vectors that get annihilated by Π_S. The space $S^\perp$ is the null space of Π_S.

Normal matrices

The matrix $A = \mathbb{R}^{n \times n}$ is *normal* if it obeys $A^{\mathsf{T}} A = A A^{\mathsf{T}}$. If A is normal, it has the following properties:

- $\vec{v}$ is an eigenvector of A if and only if $\vec{v}$ is an eigenvector of A^{T}.
- For all vectors $\vec{v}$ and $\vec{w}$ and a normal transformation A, we have

  $$(A\vec{v})\cdot(A\vec{w}) = (A\vec{v})^{\mathsf{T}}(A\vec{w}) = \vec{v}^{\mathsf{T}} A^{\mathsf{T}} A \vec{w} = \vec{v}^{\mathsf{T}} A A^{\mathsf{T}} \vec{w} = (A^{\mathsf{T}}\vec{v})\cdot(A^{\mathsf{T}}\vec{w}).$$

- The matrix A has a full set of linearly independent eigenvectors. Eigenvectors corresponding to distinct eigenvalues are orthogonal and eigenvectors from the same eigenspace can be chosen to be mutually orthogonal.

Every normal matrix is diagonalizable by an orthogonal matrix O. The eigendecomposition of a normal matrix is written as $A = O \Lambda O^{\mathsf{T}}$, where O is orthogonal and Λ is diagonal.

Orthogonal ($O^{\mathsf{T}} O = \mathbb{1}$) and symmetric ($A^{\mathsf{T}} = A$) matrices are normal matrices, since $O^{\mathsf{T}} O = \mathbb{1} = O O^{\mathsf{T}}$ and $A^{\mathsf{T}} A = A^2 = A A^{\mathsf{T}}$.

Discussion

We've defined several special categories of matrices and described their properties. You're now equipped with some very precise terminology for describing different types of matrices. Each of these special matrices plays a role in certain applications.

This section also highlighted the importance of the eigenvalue description of matrices. Indeed, we can understand all special matrices in terms of the constraints imposed on their eigenvalues. The concept map in Figure 7.1 summarizes the relationships between the different special types of matrices. The map also refers to *unitary* and *Hermitian* matrices, which extend the concepts of *orthogonal* and *symmetric* matrices to describe matrices with complex coefficients.

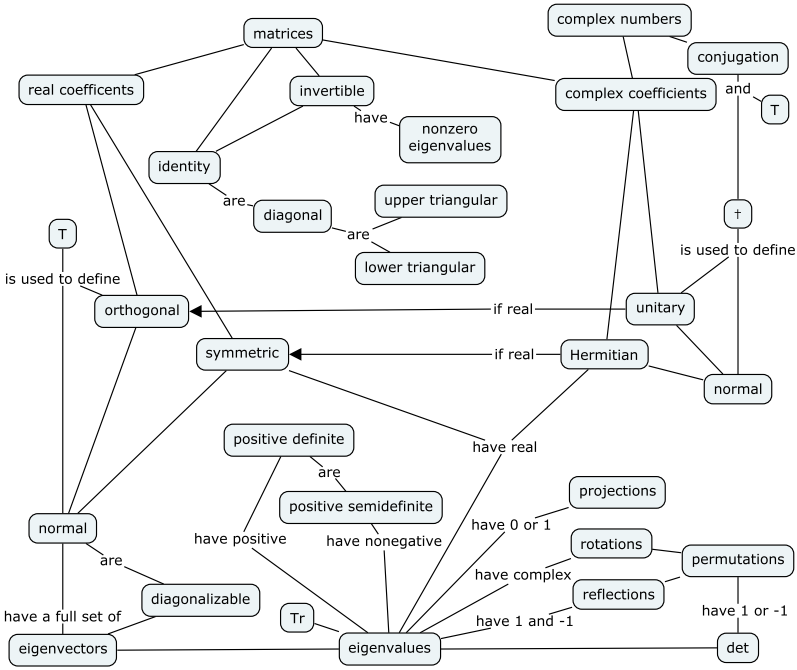


Figure 7.1: This concept map illustrates the connections and relations between special types of matrices. We can understand many special types of matrices in connection with some constraint imposed on their eigenvalues or their determinants. This diagram shows only a subset of the many connections between special matrices. Matrices with complex coefficients will be discussed in Section 7.7.

Exercises

E7.8 Is the matrix $\left[\begin{smallmatrix} 1 & 2+i \\ 3-i & 4 \end{smallmatrix}\right]$ Hermitian?

E7.9 Find the determinants and inverses of the following triangular matrices:

$$A = \begin{bmatrix} 1 & 4 & 56 \\ 0 & 5 & 14 \\ 0 & 0 & 3 \end{bmatrix}, \qquad B = \begin{bmatrix} x & 0 \\ y & z \end{bmatrix}, \qquad C = \begin{bmatrix} \frac{1}{5} & 0 \\ 0 & 5 \end{bmatrix}.$$

7.3 Abstract vector spaces

You can apply your knowledge of vectors more generally to other vector-like mathematical objects. For example, polynomials behave similarly to vectors. To add two polynomials $P(x)$ and $Q(x)$, we add together the coefficients of each power of x—the the same way vectors are added component by component.

In this section, we'll learn how to use the terminology and concepts associated with vectors to study other mathematical objects. In particular, we'll see that notions such as *linear independence*, *basis*, and *dimension* can be applied to mathematical objects like matrices, polynomials, and functions.

Definitions

An abstract vector space $(V, F, +, \cdot)$ consists of four things:

- A set of vector-like objects $V = \{\mathbf{u}, \mathbf{v}, \ldots\}$
- A field F of scalar numbers, usually $F = \mathbb{R}$
- An addition operation "+" for elements of V that dictates how to add vectors: $\mathbf{u} + \mathbf{v}$
- A scalar multiplication operation "$\cdot$" for scaling a vector by an element of the field. Scalar multiplication is usually denoted implicitly $\alpha\mathbf{u}$ (without the dot).

A vector space satisfies the following eight axioms, for all scalars $\alpha, \beta \in F$ and all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$:

1. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ (associativity of addition)
2. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ (commutativity of addition)
3. There exists a zero vector $\mathbf{0} \in V$, such that $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$ for all $\mathbf{u} \in V$.
4. For every $\mathbf{u} \in V$, there exists an inverse element $-\mathbf{u}$ such that $\mathbf{u} + (-\mathbf{u}) = \mathbf{u} - \mathbf{u} = \mathbf{0}$.

5. $\alpha(\mathbf{u}+\mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$ (distributivity I)
6. $(\alpha+\beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$ (distributivity II)
7. $\alpha(\beta\mathbf{u}) = (\alpha\beta)\mathbf{u}$ (associativity of scalar multiplication)
8. There exists a unit scalar 1 such that $1\mathbf{u} = \mathbf{u}$.

If you know anything about vectors, the above properties should be familiar. Indeed, these are the standard properties for the vector space $\mathbb{R}^n$, where the field F is $\mathbb{R}$, and for which standard vector addition and scalar multiplication operations apply.

Theory

Believe it or not, we're actually done with all the theory for this section. Move along folks, there's nothing more to see here aside from the definitions above—which are restatements of the properties of vector addition and vector scaling that you've already seen before.

The only thing left to do is illustrate these concepts through some examples.

Examples

Matrices, polynomials, and functions are vector-like math objects. The following examples demonstrate how we can treat these math objects as abstract vector spaces $(V, F, +, \cdot)$.

Matrices

Consider the vector space of $m \times n$ matrices over the real numbers $\mathbb{R}^{m\times n}$. The addition operation for two matrices $A, B \in \mathbb{R}^{m\times n}$ is the usual rule of matrix addition: $(A+B)_{ij} = a_{ij} + b_{ij}$.

This vector space is mn-dimensional, which can be seen by constructing a basis for the space. The standard basis for $\mathbb{R}^{m\times n}$ consists of matrices with zero entries everywhere except for a single 1 in the i^{th} row and the j^{th} column. Any matrix $A \in \mathbb{R}^{m\times n}$ can be written as a linear combination of the matrices in the standard basis.

Example The standard basis B_s for the vector space $\mathbb{R}^{2\times 2}$ is

$$\mathbf{e}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{e}_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Any matrix $A \in \mathbb{R}^{2\times 2}$ can be written as a linear combination of the elements of B_s:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + a_{12}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + a_{21}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + a_{22}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 + a_{21}\mathbf{e}_3 + a_{22}\mathbf{e}_4.$$

In other words, A can be expressed as a vector of coefficients with respect to the basis B_s: $A = (a_{11}, a_{12}, a_{21}, a_{22})_{B_s}$.

The abstract concept of a matrix $A \in \mathbb{R}^{2\times 2}$ can be expressed as two equivalent representations. We can think of A either as an array of coefficients with two columns and two rows, or as a four-dimensional vector of coefficients with respect to the basis B_s:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \equiv A \equiv (a_{11}, a_{12}, a_{21}, a_{22})_{B_s}.$$

We've arrived at a major *knowledge buzz* milestone: **matrices are vectors!** In precise mathematical terms, we just demonstrated the existence of an *isomorphism* between the set of 2×2 matrices and the set of four-dimensional vectors. We can add, subtract, and scale 2×2 matrices in their $\mathbb{R}^4$ representations. In the following exercises, we'll see how to compute the matrix trace operation $\mathrm{Tr}(A)$ in terms of the vector representation.

Symmetric 2x2 matrices

Define the vector space consisting of 2×2 symmetric matrices

$$\mathbb{S}(2,2) \equiv \{A \in \mathbb{R}^{2\times 2} \mid A = A^{\mathsf{T}}\}$$

in combination with the usual matrix addition and scalar multiplication operations. We obtain an explicit basis for this space as follows:

$$\mathbf{v}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Any element of the vector space $S \in \mathbb{S}(2,2)$ can be written as a linear combination of the basis elements:

$$S = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = a\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + c\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3.$$

Since there are three vectors in a basis for $\mathbb{S}(2,2)$, the vector space $\mathbb{S}(2,2)$ is three-dimensional.

Note how we count the dimensions in this case. The space of 2×2 matrices is four-dimensional in general, but imposing the symmetry constraint $a_{12} = a_{21}$ eliminates one parameter, so we're left with a three-dimensional space.

Polynomials of degree n

Define the vector space $P_n(t)$ of polynomials with real coefficients and degree less than or equal to n. The "vectors" in this space are polynomials of the form

$$\mathbf{p} = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

where $a_0, a_1, \ldots, a_n$ are the *coefficients* of the polynomial $\mathbf{p}$.

The addition of vectors $\mathbf{p}, \mathbf{q} \in P_n(t)$ is performed component-wise:

$$\begin{aligned}\mathbf{p} + \mathbf{q} &= (a_0 + a_1 x + \cdots + a_n x^n) + (b_0 + b_1 x + \cdots + b_n x^n) \\ &= (a_0 + b_0) + (a_1 + b_1)x + \cdots + (a_n + b_n)x^n.\end{aligned}$$

Similarly, scalar multiplication acts as you would expect:

$$\alpha\mathbf{p} = \alpha \cdot (a_0 + a_1 x + \ldots a_n x^n) = (\alpha a_0) + (\alpha a_1)x + \ldots (\alpha a_n)x^n.$$

The space $P_n(x)$ is $(n+1)$-dimensional since each "vector" in this space has $n+1$ coefficients.

Functions

Another interesting vector space is the set of functions $f : \mathbb{R} \to \mathbb{R}$ in combination with the point-wise addition and scaler multiplication operations:

$$\mathbf{f} + \mathbf{g} = (f + g)(x) = f(x) + g(x), \qquad \alpha\mathbf{f} = (\alpha f)(x) = \alpha f(x).$$

The space of functions is *infinite*-dimensional.

Discussion

We've talked about bases, components, and dimensions of *abstract* vector spaces. Indeed, these notions are well-defined for any vector-like object. Though this section only discussed vector spaces with real coefficients, we can apply the same techniques to vectors with coefficients from any *field*. The notion of a *field* describes any number-like object for which the operations of addition, subtraction, multiplication, and division are defined. An example of another field is the set of complex numbers $\mathbb{C}$. We'll discuss the linear algebra of vectors and matrices with complex coefficients in Section 7.7.

In the next section, we'll define an *abstract inner product* operation and use this definition to discuss concepts like orthogonality, length, and distance in abstract vector spaces.

Links

[Further discussion and examples on Wikipedia]
`http://en.wikipedia.org/wiki/Vector_space`

[Examples of vector spaces]
`http://wikibooks.org/wiki/Linear_Algebra/Definition_and_Examples_of_Vector_Spaces`

Exercises

E7.10 Consider an arbitrary matrix $A \in \mathbb{R}^{2\times 2}$ and its representation as a vector of coefficients with respect to B_s: $\vec{A} = (a_{11}, a_{12}, a_{21}, a_{22})_{B_s}$. Suppose we want to compute the matrix trace operation in terms of the vector dot product. What vector $\vec{v} \in \mathbb{R}^4$ makes this equation true $\text{Tr}(A) = \vec{v} \cdot \vec{A}$?

E7.11 Repeat the previous question, but now think of $\vec{A}$ as a 4×1 matrix. Find the matrix V that implements the trace operation: $\text{Tr}(A) = V\vec{A}$. Assume the standard matrix-matrix product is used.

E7.12 Find the dimension of the vector space of functions that satisfy $f'(t) + f(t) = 0$.

Hint: Which function is equal to a multiple of its own derivative?

E7.13 Can every polynomial of degree at most 2 be written in the form $\alpha(1) + \beta(x-1) + \gamma(x-1)^2$?

Hint: Try to express an arbitrary polynomial in this form.

7.4 Abstract inner product spaces

An inner product space is an abstract vector space $(V, \mathbb{R}, +, \cdot)$ for which we define an *abstract inner product* operation that takes pairs of vectors as inputs and produces numbers as outputs:

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}.$$

We can use any inner product operation, as long as it satisfies the following criteria for all $\mathbf{u}, \mathbf{v}, \mathbf{v}_1, \mathbf{v}_2 \in V$ and $\alpha, \beta \in \mathbb{R}$. The inner product operation must be:

- Symmetric: $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
- Linear: $\langle \mathbf{u}, \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 \rangle = \alpha\langle \mathbf{u}, \mathbf{v}_1 \rangle + \beta\langle \mathbf{u}, \mathbf{v}_2 \rangle$
- Positive semidefinite: $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ for all $\mathbf{u} \in V$ with $\langle \mathbf{u}, \mathbf{u} \rangle = 0$ if and only if $\mathbf{u} = \mathbf{0}$

These criteria are inspired by the properties of the standard inner product (dot product) for vectors in $\mathbb{R}^n$:

$$\langle \vec{u}, \vec{v} \rangle \equiv \vec{u} \cdot \vec{v} = \sum_{i=1}^{n} u_i v_i = \vec{u}^{\mathsf{T}} \vec{v}.$$

In this section, the idea of dot product is generalized to *abstract vectors* $\mathbf{u}, \mathbf{v} \in V$ by defining an inner product operation $\langle \mathbf{u}, \mathbf{v} \rangle$ appropriate for the elements of V. We'll define an inner product operation for matrices $\langle A, B \rangle$, polynomials $\langle \mathbf{p}, \mathbf{q} \rangle$, and functions $\langle f, g \rangle$. This inner product will allow us to talk about *orthogonality* between abstract vectors,

$$\mathbf{u} \text{ and } \mathbf{v} \text{ are orthogonal} \quad \Leftrightarrow \quad \langle \mathbf{u}, \mathbf{v} \rangle = 0,$$

the *length* of an abstract vector,

$$\|\mathbf{u}\| \equiv \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle},$$

and the *distance* between two abstract vectors,

$$d(\mathbf{u}, \mathbf{v}) \equiv \|\mathbf{u} - \mathbf{v}\| = \sqrt{\langle (\mathbf{u} - \mathbf{v}), (\mathbf{u} - \mathbf{v}) \rangle}.$$

Let's get started.

Definitions

We'll work with vectors from an abstract vector space $(V, \mathbb{R}, +, \cdot)$ where:

- V is the set of vectors in the vector space
- $\mathbb{R}$ is the *field* of real numbers. The coefficients of the generalized vectors are taken from this field.
- $+$ is the addition operation defined for elements of V
- $\cdot$ is the scalar multiplication operation between an element of the field $\alpha \in \mathbb{R}$ and a vector $\mathbf{u} \in V$

We define a new operation called *abstract inner product* for that space:

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}.$$

The abstract inner product takes as inputs two vectors $\mathbf{u}, \mathbf{v} \in V$ and produces real numbers as outputs: $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{R}$.

We define the following related quantities in terms of the inner product operation:

- $\|\mathbf{u}\| \equiv \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$: the *norm* or *length* of an abstract vector
- $d(\mathbf{u}, \mathbf{v}) \equiv \|\mathbf{u} - \mathbf{v}\|$: the *distance* between two vectors

Orthogonality

Recall that two vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$ are said to be orthogonal if their dot product is zero. This follows from the geometric interpretation of the dot product:

$$\vec{u} \cdot \vec{v} = \|\vec{u}\|\|\vec{v}\| \cos\theta,$$

where θ is the *angle* between $\vec{u}$ and $\vec{v}$. Orthogonal means "at right angle with." Indeed, if $\vec{u} \cdot \vec{v} = 0$, the angle between $\vec{u}$ and $\vec{v}$ must be $90°$ or $270°$, since $\cos\theta = 0$ only for these two angles.

In analogy with the regular dot product, we define the notion of *orthogonality* between abstract vectors in terms of the abstract inner product:

$$\mathbf{u} \text{ and } \mathbf{v} \text{ are orthogonal} \qquad \Leftrightarrow \qquad \langle \mathbf{u}, \mathbf{v} \rangle = 0.$$

Translating the geometrical intuition of "at $90°$ or $270°$ angle with" might not be possible for certain abstract vector spaces. For instance, what is the "angle" between two polynomials? Nevertheless, the fundamental notion of "perpendicular to" exists in all abstract inner product vector spaces.

Norm

Every definition of an inner product for an abstract vector space $(V, \mathbb{R}, +, \cdot)$ induces a *norm* for that vector space:

$$\|.\| : V \to \mathbb{R}.$$

The norm is defined in terms of the inner product. The norm of a vector is the square root of the inner product of the vector with itself:

$$\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}.$$

The norm of a vector corresponds, in some sense, to the "length" of the vector. All norms must satisfy the following criteria:

- $\|\mathbf{v}\| \geq 0$ with equality if and only if $\mathbf{v} = \mathbf{0}$
- $\|k\mathbf{v}\| = k\|\mathbf{v}\|$
- The triangle inequality:

$$\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$$

- Cauchy-Schwarz inequality:

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \, \|\mathbf{y}\|,$$

  with equality if and only if $\mathbf{x}$ and $\mathbf{y}$ are linearly dependent

Norms defined in terms of a valid inner product automatically satisfy these criteria.

Distance

The distance between two points p and q in $\mathbb{R}^n$ is equal to the norm of the vector that goes from p to q: $d(p,q) = \|q - p\|$. We can similarly define a *distance* function between pairs of vectors in an abstract vector space V:

$$d : V \times V \to \mathbb{R}.$$

The distance between two abstract vectors is equal to the norm of their difference:

$$d(\mathbf{u}, \mathbf{v}) \equiv \|\mathbf{u} - \mathbf{v}\| = \sqrt{\langle (\mathbf{u} - \mathbf{v}), (\mathbf{u} - \mathbf{v}) \rangle}.$$

Distances defined in terms of a valid norm obey the following criteria:

- $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{u})$
- $d(\mathbf{u}, \mathbf{v}) \geq 0$ with equality if and only if $\mathbf{u} = \mathbf{v}$

Examples

Let's define some inner product functions for the aforementioned abstract vector spaces.

Matrix inner product

The Hilbert–Schmidt inner product for real matrices is defined in terms of the matrix transpose, matrix product, and matrix trace operations:

$$\langle A, B \rangle_{\text{HS}} = \text{Tr}\big(A^{\mathsf{T}} B\big) .$$

We can use this inner product to talk about *orthogonality* properties of matrices. In the last section we defined the set of 2×2 symmetric matrices

$$\mathbb{S}(2,2) = \{ A \in \mathbb{R}^{2 \times 2} \mid A = A^{\mathsf{T}} \},$$

and gave an explicit basis for this space:

$$\mathbf{v}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{v}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{v}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

It's easy to show that these vectors are all *mutually orthogonal* with respect to the Hilbert–Schmidt inner product $\langle \cdot, \cdot \rangle_{\text{HS}}$:

$$\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_{\text{HS}} = 0, \qquad \langle \mathbf{v}_1, \mathbf{v}_3 \rangle_{\text{HS}} = 0, \qquad \langle \mathbf{v}_2, \mathbf{v}_3 \rangle_{\text{HS}} = 0.$$

Verify these three equations by computing each inner product. Try this by hand on a piece of paper, like right now.

The three inner product calculations of the last equation indicate that the set $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ forms an orthogonal basis for the vector space $\mathbb{S}(2,2)$ with respect to the inner product $\langle \cdot, \cdot \rangle_{\text{HS}}$.

Hilbert–Schmidt norm

The Hilbert–Schmidt inner product induces the Hilbert–Schmidt norm:

$$||A||_{\text{HS}} \equiv \sqrt{\langle A, A \rangle_{\text{HS}}} = \sqrt{\text{Tr}(A^{\mathsf{T}} A)} = \left[\sum_{i,j=1}^{n} |a_{ij}|^2 \right]^{\frac{1}{2}}.$$

We can use this norm to describe the "length" of a matrix. Continuing with the above example, we can obtain an ortho**normal** basis $\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3\}$ for $\mathbb{S}(2,2)$ as follows:

$$\hat{\mathbf{v}}_1 = \mathbf{v}_1, \qquad \hat{\mathbf{v}}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|_{\text{HS}}} = \frac{1}{\sqrt{2}} \mathbf{v}_2, \qquad \hat{\mathbf{v}}_3 = \mathbf{v}_3.$$

Verify that $\|\hat{\mathbf{v}}_2\|_{\text{HS}} = 1$.

Function inner product

Consider two functions $\mathbf{f} = f(t)$ and $\mathbf{g} = g(t)$, and define their inner product as follows:

$$\langle \mathbf{f}, \mathbf{g} \rangle \equiv \int_{-\infty}^{\infty} f(t) g(t) \, dt.$$

This formula is the continuous-variable version of the inner product formula for vectors $\vec{u} \cdot \vec{v} = \sum_i u_i v_i$. Instead of a summation, we have an integral; otherwise the idea is the same: we measure the *overlap* between $\mathbf{f}$ and $\mathbf{g}$. The integral passes along the real line from $-\infty$ until ∞ like a zipper that multiplies $f(t)$ times $g(t)$ at each point.

Example Consider the function inner product on the interval $[-1, 1]$ as defined by the formula:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_{-1}^{1} f(t) g(t) \, dt.$$

Verify that the following polynomials, known as the Legendre polynomials $P_n(x)$, are mutually orthogonal with respect to the above inner product:

$$P_0(x) = 1, \qquad P_1(x) = x,$$
$$P_2(x) = \frac{1}{2}(3x^2 - 1), \qquad P_3(x) = \frac{1}{2}(5x^3 - 3x).$$

Generalized dot product

We can think of the regular dot product for vectors as the following vector-matrix-vector product:

$$\vec{u} \cdot \vec{v} = \vec{u}^\mathsf{T}\vec{v} = \vec{u}^\mathsf{T}\mathbb{1}\vec{v}.$$

More generally, we can insert any *symmetric*, *positive semidefinite* matrix M between the vectors and obtain a valid inner product:

$$\langle \vec{x}, \vec{y} \rangle_M \equiv \vec{x}^\mathsf{T} M \vec{y}.$$

The matrix M is called the *metric* for this inner product, and it encodes the relative contributions of the different components of the vectors to the inner product.

The requirement that M be symmetric stems from the symmetric requirement for inner products: $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$. The requirement that the matrix be positive semidefinite comes from the positive semidefinite requirement for inner products: $\langle \mathbf{u}, \mathbf{u} \rangle = \vec{u}^\mathsf{T} M \vec{u} \geq 0$, for all $\mathbf{u} \in V$.

We can always obtain a symmetric and positive semidefinite matrix M by setting $M = A^\mathsf{T}A$ for some matrix A. To understand why we might want to construct M in this way, recall that each matrix A implements some linear transformation $T_A(\vec{u}) = A\vec{u}$. An inner product $\langle \vec{u}, \vec{v} \rangle_M$ can be interpreted as the regular dot product in the output space of T_A:

$$\langle \vec{u}, \vec{v} \rangle_M = \vec{u}^\mathsf{T} M \vec{v} = \vec{u}^\mathsf{T} A^\mathsf{T} A \vec{v} = (A\vec{u})^\mathsf{T}(A\vec{v}) = T_A(\vec{u}) \cdot T_A(\vec{v}).$$

This is a very powerful idea with many applications.

Example Consider the task of computing the similarity between two documents $\vec{w}_i$ and $\vec{w}_j$. Each document is represented as a vector of word counts in $\mathbb{R}^n$. The first component of $\vec{w}_i$ represents how many times the word `aardvark` appears in document i, the second component is the count of `abacus`, and so on for the n words in the dictionary.

Intuitively, two vectors are similar if their inner product is large. We can compute the similarity between documents $\vec{w}_i$ and $\vec{w}_j$ by calculating their inner product, normalized by their norms:

$$\text{sim}_{\mathbb{1}}(\vec{w}_i, \vec{w}_j) \equiv \frac{\langle \vec{w}_i, \vec{w}_j \rangle}{\|\vec{w}_i\|\|\vec{w}_j\|} = \frac{\vec{w}_i^\mathsf{T}\mathbb{1}\vec{w}_j}{\|\vec{w}_i\|\|\vec{w}_j\|}.$$

This expression is called the *cosine similarity* between vectors because it corresponds to the cosine of the angle between the vectors

$\vec{w}_i$ and $\vec{w}_j$. To make things more concrete, suppose we use a vocabulary of 40 000 words to represent documents, making the word count vectors for each document 40 000-dimensional vectors. That's a lot of dimensions! Using the function $\text{sim}_{\mathbb{1}}$ to compute similarities between documents won't work too well because it's pretty rare that two vectors in a 40 000-dimensional space point in the same direction.

Consider now the linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ that transforms the word vectors into "topic" vectors, which are m-dimensional vectors with $m < n$. Each dimension in "topic space" corresponds to some topic: `art`, `literature`, `science`, etc., and we can assume there are a small number of topics. Using T, we can obtain the topic representation of each document $\vec{\theta}_i = T(\vec{w}_i)$, where the components of $\vec{\theta}_i \in \mathbb{R}^m$ tell us the proportions of the different topics that appear in document i.

An improved similarity measure for documents is to compute the cosine similarity between their vector representations in topics space:

$$\text{sim}_T(\vec{w}_i, \vec{w}_j) \equiv \frac{\langle \vec{\theta}_i, \vec{\theta}_j \rangle}{\|\vec{\theta}_i\| \|\vec{\theta}_j\|} = \frac{\langle T(\vec{w}_i), T(\vec{w}_j) \rangle}{\|T(\vec{w}_i)\| \|T(\vec{w}_j)\|}.$$

In words, sim_T measures the cosine similarity between documents based on the topics they contain. Assuming we pick a small number of topics, say $m = 40$, the similarity metric sim_T will lead to improved similarity computations because the inner product will be computed in a smaller vector space.

Let's focus on the numerator of the expression for sim_T. Recall that every linear transformation T can be represented as a matrix-vector product $T(\vec{w}_i) = A_T \vec{w}_i$, for some matrix A_T. We can write

$$\langle T(\vec{w}_i), T(\vec{w}_j) \rangle = \langle A_T \vec{w}_i, A_T \vec{w}_j \rangle = \vec{w}_i^{\mathsf{T}} \underbrace{A_T^{\mathsf{T}} A_T}_{M} \vec{w}_j \equiv \langle \vec{w}_i, \vec{w}_j \rangle_M,$$

which shows the inner product in topic-space can be interpreted as a *generalized inner product* for word-vectors with metric $M = A_T^{\mathsf{T}} A_T$.

The notion of a generalized inner product with metric matrix M defined via $\langle \vec{u}, \vec{v} \rangle_M \equiv \vec{u}^{\mathsf{T}} M \vec{v}$ is an important concept that appears in many advanced math topics like analysis and differential geometry. Metrics also shows up in physics: when Einstein talks about how masses cause space to become "curved," he's really talking about the curvature of the metric of space-time.

Valid and invalid inner product spaces

A standard question profs like to ask on exams is to check whether a given vector space and some weird definition of an inner product

operation form a valid inner product space. Recall that *any* operation can be used as the inner product, as long as it satisfies the *symmetry*, *linearity*, and *positive semidefinite* criteria. To prove an inner product operations is valid, you must show it satisfies the three criteria.

Alternatively, you can prove the vector space $(V, \mathbb{R}, +, \cdot)$ with inner product $\langle \mathbf{u}, \mathbf{v} \rangle$ is *not* a valid inner product space if you find an example of one of more $\mathbf{u}, \mathbf{v} \in V$ which do not satisfy one of the axioms.

Discussion

This has been another one of those sections where we learn no new linear algebra, but simply generalize notions we already know about standard vectors $\vec{v} \in \mathbb{R}^n$ to abstract vector-like objects $\mathbf{v} \in V$. You can now talk about orthogonality and norms for matrices, polynomials, and functions.

Exercises

7.5 Gram–Schmidt orthogonalization

Recall what we learned in Section 5.3 about the three "quality grades" for bases: orthonormal, orthogonal, and generic, with orthonormal bases being the most desirable of the three. In this section, we'll learn how to take a generic basis for an n-dimensional space V—that is, a set of n linearly independent vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$—and transform it into an orthonormal basis $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}$ that obeys:

$$\langle \hat{\mathbf{e}}_i, \hat{\mathbf{e}}_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

This procedure is known as *Gram–Schmidt orthogonalization* and is based on a sequence of projection and subtraction operations.

The discussion and procedures in this section are described in terms of vectors in an abstract inner product space. Thus, the Gram–Schmidt algorithm applies to ordinary vectors $\vec{v} \in \mathbb{R}^n$, matrices $A \in \mathbb{R}^{m \times n}$, and polynomials $\mathbf{p} \in P_n(x)$. Indeed, we can talk about orthogonality for any set of mathematical objects for which we've defined an inner product operation.

Definitions

- V: an n-dimensional vector space
- $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$: a generic basis for the space V
- $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$: an *orthogonal basis* for V. Each vector $\mathbf{e}_i$ is orthogonal to all other vectors: $\mathbf{e}_i \cdot \mathbf{e}_j = 0$, for $i \neq j$.

- $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}$: an *orthonormal* basis for V. An orthonormal basis is an orthogonal basis of unit-length vectors.

We assume the vector space V is equipped with an inner product operation:

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}.$$

The following operations are defined in terms of the inner product:

- The *length* of a vector $\|\mathbf{v}\| = \langle \mathbf{v}, \mathbf{v} \rangle$
- The *projection* operation. The projection of the vector $\mathbf{u}$ onto the subspace spanned by the vector $\mathbf{e}$ is denoted $\Pi_{\mathbf{e}}(\mathbf{u})$ and computed using
$$\Pi_{\mathbf{e}}(\mathbf{u}) = \frac{\langle \mathbf{u}, \mathbf{e} \rangle}{\|\mathbf{e}\|^2} \mathbf{e}.$$
- The *projection complement* of the projection $\Pi_{\mathbf{e}}(\mathbf{u})$ is the vector $\mathbf{w}$ that we must add to $\Pi_{\mathbf{e}}(\mathbf{u})$ to recover the original vector $\mathbf{u}$:
$$\mathbf{u} = \Pi_{\mathbf{e}}(\mathbf{u}) + \mathbf{w} \qquad \Rightarrow \qquad \mathbf{w} = \mathbf{u} - \Pi_{\mathbf{e}}(\mathbf{u}).$$
The vector $\mathbf{w}$ is orthogonal to the vector $\mathbf{e}$, $\langle \mathbf{w}, \mathbf{e} \rangle = 0$.

Orthonormal bases are nice

Recall that a *basis* for an n-dimensional vector space V is any set of n linearly independent vectors in V. The choice of basis is a big deal because we express the components of vectors and matrices with respect to the basis. From a theoretical standpoint, all bases are equally good; but from a practical standpoint, orthogonal and orthonormal bases are much easier to work with.

An orthonormal basis $B = \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ is the most useful kind of basis because the coefficients c_1, c_2, and c_3 of a vector $\mathbf{c} = (c_1, c_2, c_3)_B$ with respect to B are obtained using three independent inner-product calculations:

$$c_1 = \langle \mathbf{c}, \hat{\mathbf{e}}_1 \rangle, \qquad c_2 = \langle \mathbf{c}, \hat{\mathbf{e}}_2 \rangle, \qquad c_3 = \langle \mathbf{c}, \hat{\mathbf{e}}_3 \rangle.$$

We can express any vector $\mathbf{v}$ as follows:

$$\mathbf{v} = \langle \mathbf{v}, \hat{\mathbf{e}}_1 \rangle \hat{\mathbf{e}}_1 + \langle \mathbf{v}, \hat{\mathbf{e}}_2 \rangle \hat{\mathbf{e}}_2 + \langle \mathbf{v}, \hat{\mathbf{e}}_3 \rangle \hat{\mathbf{e}}_3.$$

This formula is a generalization of the usual formula for coefficients with respect to the standard basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$: $\vec{v} = (\vec{v} \cdot \hat{\imath})\hat{\imath} + (\vec{v} \cdot \hat{\jmath})\hat{\jmath} + (\vec{v} \cdot \hat{k})\hat{k}$.

Orthogonalization

The "best" kind of basis for computational purposes is an orthonormal basis like $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}$. How can we *upgrade* some general set of n linearly independent vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ into an orthonormal basis $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}$? The vectors $\{\hat{\mathbf{e}}_i\}$ must be linear combinations of the vectors $\{\mathbf{v}_i\}$, but which linear combinations should we choose?

Note the vector space V remains the same:

$$\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} = V = \text{span}\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}.$$

However, the basis $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_m\}$ is easier to work with.

The technical term for distilling a high-quality orthonormal basis from a low-quality basis of arbitrary vectors is called *orthogonalization*. Most of the work lies in obtaining the set of vectors $\{\mathbf{e}_i\}$ that are *orthogonal* to each other:

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 0, \quad \text{for all } i \neq j.$$

To convert an orthogonal basis into an orthonormal basis, divide each vector by its length: $\hat{\mathbf{e}}_i = \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|}$.

It's now time to see how orthogonalization works; get ready for some Gram–Schmidting.

Gram–Schmidt orthogonalization procedure

The Gram–Schmidt orthogonalization procedure converts a basis of arbitrary vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ into an orthonormal basis $\{\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_n\}$. The main idea is to take the vectors $\mathbf{v}_i$ one at a time, each time defining a new vector $\mathbf{e}_i$ as the *orthogonal complement* of $\mathbf{v}_i$ to all the previously chosen vectors $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_{i-1}$. Recall we can use the projection formula $\Pi_{\hat{\mathbf{e}}}(\mathbf{v}) \equiv \langle \hat{\mathbf{e}}, \mathbf{v} \rangle \hat{\mathbf{e}}$ to compute the component of any vector $\mathbf{v}$ in the direction $\hat{\mathbf{e}}$.

The orthogonalization algorithm consists of n steps:

$$\begin{aligned}
\mathbf{e}_1 &= \mathbf{v}_1 & \hat{\mathbf{e}}_1 &= \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \\
\mathbf{e}_2 &= \mathbf{v}_2 - \Pi_{\hat{\mathbf{e}}_1}(\mathbf{v}_2), & \hat{\mathbf{e}}_2 &= \frac{\mathbf{e}_2}{\|\mathbf{e}_2\|}, \\
\mathbf{e}_3 &= \mathbf{v}_3 - \Pi_{\hat{\mathbf{e}}_1}(\mathbf{v}_3) - \Pi_{\hat{\mathbf{e}}_2}(\mathbf{v}_3), & \hat{\mathbf{e}}_3 &= \frac{\mathbf{e}_3}{\|\mathbf{e}_3\|}, \\
\mathbf{e}_4 &= \mathbf{v}_4 - \Pi_{\hat{\mathbf{e}}_1}(\mathbf{v}_4) - \Pi_{\hat{\mathbf{e}}_2}(\mathbf{v}_4), -\Pi_{\hat{\mathbf{e}}_3}(\mathbf{v}_4), & \hat{\mathbf{e}}_4 &= \frac{\mathbf{e}_4}{\|\mathbf{e}_4\|}, \\
&\vdots & &\vdots \\
\mathbf{e}_n &= \mathbf{v}_n - \sum_{i=1}^{n-1} \Pi_{\hat{\mathbf{e}}_i}(\mathbf{v}_n), & \hat{\mathbf{e}}_n &= \frac{\mathbf{e}_n}{\|\mathbf{e}_n\|}.
\end{aligned}$$

In the j^{th} step of the procedure, we compute a vector $\mathbf{e}_j$ by starting from $\mathbf{v}_j$ and subtracting all the projections of $\mathbf{v}_j$ onto the previous vectors $\mathbf{e}_i$ for all $i < j$. In other words, $\mathbf{e}_j$ is the part of $\mathbf{v}_j$ that is orthogonal to all the vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{j-1}$.

This procedure is known as orthogonalization because it splits the vector space V into orthogonal subspaces $V_1, V_2, \ldots, V_n$:

$$V_j = \text{span}\left\{ \mathbf{v} \in V \mid \mathbf{v} = \sum_{i=1}^{j} \alpha_i \mathbf{v}_i \right\} \setminus \text{span}\left\{ \mathbf{v} \in V \mid \mathbf{v} = \sum_{i=1}^{j-1} \alpha_i \mathbf{v}_i \right\}.$$

Recall that the symbol $\setminus$ denotes the *set minus* operation. The set $A \setminus B$ consists of all elements that are in set A but not in set B.

Observe the subspaces $V_1, V_2, \ldots, V_n$ are, by construction, mutually orthogonal. Given any vector $\mathbf{u} \in V_i$ and another vector $\mathbf{v} \in V_j, j \neq i$, then $\mathbf{u} \cdot \mathbf{v} = 0$.

The vector space V is the sum of these subspaces:

$$V = V_1 \oplus V_2 \oplus V_3 \oplus \cdots \oplus V_n.$$

The notation $\oplus$ means *orthogonal sum*. Each space V_j is spanned by a vector $\mathbf{e}_j$ which is orthogonal to all the V_is, for $i < j$.

Discussion

The main point you must remember about orthogonalization is simply that it can be done. Any "low-quality" basis (a set of n linearly independent vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ in an n-dimensional space) can be converted into a "high quality" orthonormal basis $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_n\}$ using the Gram–Schmidt procedure.

You can also perceive the Gram–Schmidt procedure as a technique for creating structure in an arbitrary vector space V. The initial description $V = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ lacks structure. It's just some amorphous vector space spanned by an arbitrary set of vectors. After the orthogonalization procedure, we obtain the equivalent description $V = V_1 \oplus V_2 \oplus V_3 \oplus \cdots \oplus V_n$ that shows V is the direct sum of orthogonal subspaces.

In the next section, we'll continue on our mathematical quest for structure by discussing procedures that uncover hidden structure in matrices. For example, when phrased in terms of matrices, the Gram–Schmidt orthogonalization procedure is called *QR decomposition*—stay tuned! And meanwhile, try the following exercises.

Exercises

E7.14 Convert the vectors $\vec{v}_1 = (4, 2)$ and $\vec{v}_2 = (1, 3)$ into an orthogonal basis for $\mathbb{R}^2$.

E7.15 Perform the Gram–Schmidt orthogonalization procedure on the vectors $\vec{v}_1 = (1,1,0)$, $\vec{v}_2 = (1,0,1)$, and $\vec{v}_3 = (0,1,1)$ to obtain an orthonormal basis $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$.

E7.16 Consider the vector space $P_2(x)$ of polynomials of degree at most two in combination with the inner product $\langle \mathbf{f}, \mathbf{g} \rangle \equiv \int_{-1}^{1} f(x)g(x)\,dx$. The functions $\mathbf{f}_1(x) = 1$, $\mathbf{f}_2(x) = x$, and $\mathbf{f}_3(x) = x^2$ are linearly independent and form a basis for $P_2(x)$. Find an orthonormal basis for $P_2(x)$.

7.6 Matrix decompositions

It's often useful to express a given matrix as the product of other, simpler matrices. These matrix decompositions (also known as factorizations) can help us understand the structure of matrices by revealing their constituents. In this section, we'll discuss various matrix factorizations and specify the types of matrices they apply to.

Most of the material covered here is not usually included in a first-year linear algebra course. Nevertheless, knowing about the different matrix decompositions is quite helpful, as many linear algebra applications depend on these decompositions. Got that? Good. Onward!

Eigendecomposition

The eigendecomposition breaks a matrix into its eigenvalues and eigenvectors. The eigenbasis, when it exists, is the most "natural" basis for looking at a matrix. A diagonalizable matrix A can be written as

$$A = Q\Lambda Q^{-1},$$

where Q is a matrix whose columns are eigenvectors of A, and Λ is a diagonal matrix containing the eigenvalues of A.

The eigendecomposition of a matrix is a similarity transformation (a change of basis) where the new basis matrix consists of eigenvectors of the matrix.

If A is positive semidefinite then its eigenvalues are nonnegative. If the matrix A is symmetric then its eigenvalues are real numbers.

When the matrix A is *normal*, meaning it satisfies $AA^{\mathsf{T}} = A^{\mathsf{T}}A$, we can choose Q to be an orthogonal matrix O that satisfies $O^{\mathsf{T}}O = \mathbb{1}$. Calculating the inverse of an orthogonal matrix is easy: $O^{-1} = O^{\mathsf{T}}$. The eigendecomposition for normal matrices is $A = O\Lambda O^{\mathsf{T}}$.

Singular value decomposition

We can generalize the concepts of eigenvalues and eigenvectors to non-square matrices. Consider a matrix $A \in \mathbb{R}^{m\times n}$. Since the matrix A is

not a square matrix, we can't use the standard eigendecomposition. However, there is a trick for turning a non-square matrix into a square matrix while preserving some of its properties: multiply the matrix by its transpose. The matrix $AA^T \in \mathbb{R}^{n \times n}$ has the same column space as the matrix A. Similarly, $A^TA \in \mathbb{R}^{m \times m}$ has the same row space as the matrix A.

The *singular value decomposition* breaks a matrix into the product of three matrices: an $m \times m$ orthogonal matrix U which consists of *left singular vectors*, an $m \times n$ matrix Σ with the *singular values* σ_i on the diagonal, and an $n \times n$ orthogonal matrix V^T of *right singular vectors*:

$$A = \underbrace{\begin{bmatrix} | & & | \\ \hat{u}_1 & \cdots & \hat{u}_m \\ | & & | \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ 0 & 0 & \cdots \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} — & \hat{v}_1 & — \\ & \vdots & \\ — & \hat{v}_n & — \end{bmatrix}}_{V^T} = U\Sigma V^T.$$

To find the matrices U, Σ, and V, perform eigendecomposition on the matrix products AA^T and A^TA.

First, consider first the matrix AA^T. Since AA^T is a square matrix, we can compute its eigendecomposition $AA^T = U\Lambda_\ell U^T$. The eigenvectors of AA^T span the same space as the column space of the matrix A. We call these vectors the *left singular vectors* of A.

The left singular vectors of A (the columns of U) are the eigenvectors of the matrix AA^T:

$$U = \begin{bmatrix} | & & | \\ \hat{u}_1 & \cdots & \hat{u}_m \\ | & & | \end{bmatrix}, \quad \text{where } \{(\lambda_i, \hat{u}_i)\} = \text{eigenvects}(AA^T).$$

To find the right singular vectors of A (the rows of V^T), perform the eigendecomposition on the matrix A^TA, denoted $A^TA = V\Lambda_r V^T$. Build the orthogonal matrix V^T by stacking the eigenvectors of A^TA as rows:

$$V^T = \begin{bmatrix} — & \hat{v}_1 & — \\ & \vdots & \\ — & \hat{v}_n & — \end{bmatrix}, \quad \text{where } \{(\lambda_i, \hat{v}_i)\} = \text{eigenvects}(A^TA).$$

The eigenvalues of the matrix A^TA are the same as the eigenvalues of the matrix AA^T. In both cases, the eigenvalues λ_i correspond to the squares of the singular values of the matrix A.

On its diagonal, the matrix of singular values $\Sigma \in \mathbb{R}^{m\times n}$ contains the singular values σ_i, which are the positive square roots of the eigenvalues λ_i of the matrix AA^T (or the matrix A^TA):

$$\sigma_i = \sqrt{\lambda_i}\,, \quad \text{where } \{\lambda_i\} = \text{eigenvals}(AA^T) = \text{eigenvals}(A^TA).$$

The singular value decomposition shows the inner structure of the matrix A. We can interpret the operation $\vec{y} = A\vec{x} = U\Sigma V^T\vec{x}$ as a three-step process:

1. Convert the input $\vec{x}$ to the basis of right singular vectors $\{\vec{v}_i\}$.
2. Scale each component by the corresponding singular value σ_i.
3. Convert the output from the $\{\vec{u}_i\}$ basis to the standard basis.

This three-step procedure is analogous to the three-step procedure we used to understand the eigendecomposition of square matrices in Section 7.1 (see page 264).

The singular value decomposition (SVD) has numerous applications in statistics, machine learning, and computer science. Applying the SVD to a matrix is like looking inside it with X-ray vision, since you can see its σ_is. The action of $A = U\Sigma V^T$ occurs in n parallel streams: the i^{th} stream consists of multiplying the input vector by the right singular vector $\vec{v}_i^T$, scaling by the weight σ_i, and finally multiplying by the left singular vector $\vec{u}_i$. Each singular value σ_i corresponds to the "strength" of A on the i^{th} subspace—the subspace spanned by its i^{th} left and right singular vectors.

Example Suppose you need to calculate the product $M\vec{v}$ where $M \in \mathbb{R}^{1000\times 2000}$ and $\vec{v} \in \mathbb{R}^{2000}$. Suppose furthermore the matrix M has only three large singular values, $\sigma_1 = 6$, $\sigma_2 = 5$, $\sigma_3 = 4$, and many small singular values, $\sigma_4 = 0.0002, \sigma_5 = 0.0001, \ldots, \sigma_{1000} = 1.1 \times 10^{-13}$. Observe that most of the "weight" of the matrix Σ is concentrated in the first three singular values, σ_1, σ_2, and σ_3. We can obtain a *low-rank approximation* to the matrix M by keeping only the large singular values and their associated singular vectors. Construct the matrix $\tilde{\Sigma} \in \mathbb{R}^{3\times 3}$ which contains only the first three singular values, and surround $\tilde{\Sigma}$ with matrices $\tilde{U} \in \mathbb{R}^{1000\times 3}$ and $\tilde{V}^\dagger \in \mathbb{R}^{3\times 2000}$ that contain the singular vectors associated with the first three singular values. Despite the significant reduction in the size of the matrices used in the decomposition, the matrix $\tilde{M} \equiv \tilde{U}\tilde{\Sigma}\tilde{V}^\dagger \in \mathbb{R}^{1000\times 2000}$ represents a good approximation to the original matrix M. We cut some small insignificant singular values, which didn't change the matrix too much. We can quantify the difference between

the original M and its low-rank approximation $\tilde{M}$ using the Hilbert–Schmidt norm: $||M - \tilde{M}||_{\text{HS}} = \sqrt{\sum_{i=4}^{1000} \sigma_i^2}$. Since $\sigma_4, \sigma_5, \ldots, \sigma_{1000}$ are tiny numbers, we can say $\tilde{M} \approx M$.

Links

[Singular value decomposition on Wikipedia]
`http://en.wikipedia.org/wiki/Singular_value_decomposition`

[Principal component analysis in statistics is based on SVD]
`http://en.wikipedia.org/wiki/Principal_component_analysis`

[Understanding the SVD and its applications]
`http://www.math.umn.edu/~lerman/math5467/svd.pdf`

LU decomposition

Computing the inverse of a triangular matrix is far easier than computing the inverse of a general matrix. Thus, it's sometimes useful to write a matrix as the product of two triangular matrices for computational purposes. We call this factorization the LU decomposition:

$$A = LU,$$

where U is an *upper triangular* matrix and L is a *lower triangular* matrix.

The main application of this decomposition is to obtain more efficient solutions to equations of the form $A\vec{x} = \vec{b}$. Because $A = LU$, we can solve this equation in two steps. Starting from $LU\vec{x} = \vec{b}$, first multiply by L^{-1} and then by U^{-1}:

$$LU\vec{x} = \vec{b} \quad \Rightarrow \quad L^{-1}LU\vec{x} = U\vec{x} = L^{-1}\vec{b} \quad \Rightarrow \quad \vec{x} = U^{-1}L^{-1}\vec{b}.$$

We've split the work of finding the inverse A^{-1} into two simpler subtasks: finding L^{-1} and U^{-1}, which are easier to compute.

The LU decomposition is mainly used for linear algebra calculations on computers, but it's also possible to find the LU decomposition of a matrix by hand. Recall the algorithm for finding the inverse of a matrix in which we start from the array $[\, A \,|\, \mathbb{1} \,]$ and perform row operations until we bring the array into reduced row echelon form $[\, \mathbb{1} \,|\, A^{-1} \,]$. Consider the midpoint of the algorithm when the left-hand side of the array is the row echelon form (REF). Since the matrix A in its REF is upper triangular, the array will contain $[\, U \,|\, L^{-1} \,]$. The U part of the decomposition is on the left-hand side, and the L part is obtained by finding the inverse of the right-hand side of the array.

Note the LU decomposition exists only for matrices that can be brought to RREF without using row-swap operations. If a matrix A

requires row-swap operations to be transformed to RREF, we can decompose it as $A = PLU$, where P is a permutation matrix. The PLU decomposition has the same computational advantages of splitting the inverse computation into three simpler subtasks: $A^{-1} = U^{-1}L^{-1}P^{-1}$.

Cholesky decomposition

For a *symmetric*, *positive semidefinite* matrix A, the LU decomposition can take on a simpler form. Such matrices can be written as the product of a triangular matrix and its transpose:

$$A = LL^{\mathsf{T}} \quad \text{or} \quad A = U^{\mathsf{T}}U,$$

where U is an *upper triangular* matrix and L is a *lower triangular* matrix. This is called the *Cholesky decomposition* of a matrix, and like the LU decomposition, it has applications for faster numerical linear algebra calculations, non-linear optimization, and machine learning.

QR decomposition

Any real square matrix $A \in \mathbb{R}^{n \times n}$ can be decomposed as a product of an orthogonal matrix O and an upper triangular matrix U:

$$A = OU.$$

For historical reasons, the orthogonal matrix is denoted Q instead of O, and the upper triangular matrix is denoted R (think "right-triangular" since it contains entries only to the *right* of main diagonal). Using the conventional names, the decomposition becomes

$$A = QR,$$

which is why it's known as the QR decomposition.

The QR decomposition is equivalent to the Gram–Schmidt orthogonalization procedure on the columns of the matrix. The matrix Q records the orthonormal basis while the matrix R contains the coefficients required to express the columns of A as linear combinations of the columns of Q.

Example Consider the decomposition

$$A = \begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{bmatrix} = QR.$$

We're looking for an orthogonal matrix Q and an upper triangular matrix R such that $A = QR$. We can obtain the orthogonal matrix Q by performing the Gram–Schmidt procedure on the columns of A.

Let's illustrate the procedure by computing the factorization A. Begin by changing the second column in A so it becomes orthogonal to the first (by subtracting a multiple of the first column). Next, change the third column in A so it is orthogonal to both of the first columns (by subtracting multiples of the first two columns). We obtain a matrix with the same column space as A but which has orthogonal columns:

$$\begin{bmatrix} | & | & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} 12 & -69 & -\frac{58}{5} \\ 6 & 158 & \frac{6}{5} \\ -4 & 30 & -33 \end{bmatrix}.$$

To obtain an orthogonal matrix, we must normalize each column to be of unit length:

$$Q = \begin{bmatrix} | & | & | \\ \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} & \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} & \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\ | & | & | \end{bmatrix} = \begin{bmatrix} \frac{6}{7} & -\frac{69}{175} & -\frac{58}{175} \\ \frac{3}{7} & \frac{158}{175} & \frac{6}{175} \\ -\frac{2}{7} & \frac{6}{35} & -\frac{33}{35} \end{bmatrix}.$$

We can obtain the matrix R from Q^T and A:

$$Q^\mathsf{T}A = \underbrace{Q^\mathsf{T}Q}_{\mathbb{1}}R = R \qquad \Rightarrow \qquad R = Q^\mathsf{T}A = \begin{bmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{bmatrix}.$$

The columns of R contain the mixture of coefficients required to obtain the columns of A from the columns of Q. For example, the second column of A is equal to $21\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} + 175\frac{\mathbf{u}_2}{\|\mathbf{u}_2\|}$. Verify that QR equals A.

Discussion

The last several pages have only scratched the surface of matrix decompositions. There are countless applications for matrix methods, and matrix factorizations play key roles in many of them.

Machine learning techniques often use matrix decompositions to uncovers useful structure within data matrices. Two examples include *nonnegative matrix factorization* (used for recommender systems) and *latent Dirichlet allocation* (used for document classification). I encourage you to research this subject further on your own—it's quite an interesting wormhole to get sucked into.

Links

[Cool retro video showing the steps of the SVD procedure]
`http://www.youtube.com/watch?v=R9UoFyqJca8`

[More info and examples on Wikipedia]
http://en.wikipedia.org/wiki/Matrix_decomposition
http://en.wikipedia.org/wiki/Cholesky_decomposition

[A detailed example of the QR factorization of a matrix]
http://www.math.ucla.edu/~yanovsky/Teaching/Math151B/handouts/GramSchmidt.pdf

Exercises

E7.17 Compute the QR factorization of the matrix $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$.

7.7 Linear algebra with complex numbers

So far we've discussed the math of vectors and matrices with real coefficients. In fact, the linear algebra techniques you learned apply to any *field* F. The term *field* applies to any mathematical object for which the operations of addition, subtraction, multiplication, and division are defined.

Since the complex numbers $\mathbb{C}$ are a field, we can perform linear algebra over the field of complex numbers. In this section, we'll define vectors and matrices with complex coefficients, and discover that they behave similarly to their real counterparts. You'll see that complex linear algebra is no more complex than real linear algebra: it's the same, in fact, except for one small difference: instead of matrix transpose A^T, we use the Hermitian transpose $A^\dagger$, which is the combination of the transpose and an entry-wise complex conjugate operation.

Complex vectors are not just an esoteric mathematical concept intended for specialists. Complex vectors can arise as answers for problems involving ordinary real matrices. For example, the rotation matrix

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

has complex eigenvalues $\lambda_1 = e^{i\theta}$ and $\lambda_2 = e^{-i\theta}$ and its eigenvectors have complex coefficients. If you want to know how to calculate the eigenvalues and eigenvectors of rotation matrices, you need to understand how to do linear algebra calculations with complex numbers.

This section serves as a review of all the important linear algebra concepts we've learned in this book. I recommend you read this section, even if you're not required to know about complex matrices for your course. As your guide through the land of linear algebra, it's my duty to make sure you understand linear algebra in the complex field. It's good stuff; I guarantee there's *knowledge buzz* to be had in this section.

Definitions

Recall the basic notions of complex numbers introduced in Section 2.4:

- i: the unit imaginary number; $i \equiv \sqrt{-1}$ or $i^2 = -1$
- $z = a + bi$: a complex number z whose real part is a and whose imaginary part is b
- $\mathbb{C}$: the set of complex numbers $\mathbb{C} = \{a + bi \mid a, b \in \mathbb{R}\}$
- $\text{Re}\{z\} = a$: the *real* part of $z = a + bi$
- $\text{Im}\{z\} = b$: the *imaginary* part of $z = a + bi$
- $\bar{z}$: the *complex conjugate* of z. If $z = a + bi$ then $\bar{z} = a - bi$
- $|z| = \sqrt{\bar{z}z} = \sqrt{a^2 + b^2}$: the *magnitude* or *length* of $z = a + bi$
- $\arg(z) =^1 \tan^{-1}(\frac{b}{a})$: the *phase* or *argument* of $z = a + bi$

Complex vectors

A complex vector $\vec{v} \in \mathbb{C}^n$ is an array of n complex numbers:

$$\vec{v} = (v_1, v_2, \ldots, v_n) \ \in \ (\mathbb{C}, \mathbb{C}, \ldots, \mathbb{C}) \equiv \mathbb{C}^n.$$

Complex matrices

A complex matrix $A \in \mathbb{C}^{m \times n}$ is a table of numbers with m rows and n columns. An example of a 3×2 matrix with complex entries is

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \ \in \ \begin{bmatrix} \mathbb{C} & \mathbb{C} \\ \mathbb{C} & \mathbb{C} \\ \mathbb{C} & \mathbb{C} \end{bmatrix} \equiv \mathbb{C}^{3 \times 2}.$$

Hermitian transpose

The *Hermitian transpose* operation, denoted †, consists of the combination of the regular transpose ($A \to A^\mathsf{T}$) and the complex conjugation of each entry in the matrix ($a_{ij} \to \overline{a_{ij}}$):

$$A^\dagger \equiv \overline{(A^\mathsf{T})} = (\overline{A})^\mathsf{T}.$$

Expressed in terms of the entries of the matrix a_{ij}, the Hermitian transpose corresponds to the transformation $a_{ij} \to \overline{a_{ji}}$. There are *many* mathematical terms that refer to this operation, including *Hermitian conjugate*, *complex transpose*, "dagger" operation, *conjugate transpose*, and *adjoint*.

[1] Note that $\tan^{-1}(\frac{b}{a})$ and $\arg(z)$ coincide only if $a \geq 0$, and a manual correction is necessary to the output of $\tan^{-1}(\frac{b}{a})$ when $a < 0$. Alternatively, we can use the function `atan2(b,a)` that computes the correct phase for all $z = a + bi$.

The term *adjoint* is preferred by mathematicians and the notation A^* is used consistently in mathematics research papers. The dagger notation † is preferred by physicists and engineers, but shunned by mathematicians. Mathematicians prefer to stick with the star superscript because they feel they invented the concept. We use the notation † in this book because at some point the author had to make an allegiance with one of the two camps, and because the symbol † looks a bit like the transpose symbol $^\mathsf{T}$.

The Hermitian transpose applied to a 3×2 matrix acts as follows:

$$\text{if } A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad \text{then} \quad A^\dagger = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \overline{a_{31}} \\ \overline{a_{12}} & \overline{a_{22}} & \overline{a_{32}} \end{bmatrix}.$$

Recall that vectors are special types of a matrices. We can identify a vector $\vec{v} \in \mathbb{C}^n$ with a column matrix $\vec{v} \in \mathbb{C}^{n \times 1}$ or with a row matrix $\vec{v} \in \mathbb{C}^{1 \times n}$. We apply the complex conjugation operation to transform column vectors into conjugate row vectors:

$$\vec{v}^\dagger \equiv \overline{(\vec{v}^\mathsf{T})} = (\overline{\vec{v}})^\mathsf{T}.$$

The Hermitian transpose of a column vector is a row vector in which each coefficient has been complex-conjugated:

$$\text{if } \vec{v} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad \text{then} \quad \vec{v}^\dagger = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}^\dagger = \begin{bmatrix} \overline{\alpha} & \overline{\beta} & \overline{\gamma} \end{bmatrix}.$$

The Hermitian transpose for vectors is important because it's related to the definition of the inner product for complex vectors.

Complex inner product

The inner product for vectors with complex coefficients $\vec{u}, \vec{v} \in \mathbb{C}^n$ is defined as the following operation:

$$\langle \vec{u}, \vec{v} \rangle \equiv \sum_{i=1}^{n} \overline{u_i} v_i \equiv \vec{u}^\dagger \vec{v}.$$

In this expression, complex conjugation is applied to each of the first vector's components. This corresponds to the notion of applying the Hermitian transpose to the first vector to turn it into a row vector of complex conjugates, then using the matrix multiplication rule for a $1 \times n$ matrix $\vec{u}^\dagger$ times an $n \times 1$ matrix $\vec{v}$.

For real vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$, the complex inner product formula reduces to the inner product formula we used previously: $\vec{u} \cdot \vec{v} = \vec{u}^\mathsf{T} \vec{v}$.

Rather than thinking of the inner product for complex vectors as a new operation, we can say the inner product has always been defined as $\langle \vec{u}, \vec{v} \rangle \equiv \vec{u}^\dagger \vec{v}$—we just never noticed until now because complex conjugation has no effect on vectors with real coefficients. Specifically, $\vec{u}^\dagger = \vec{u}^\mathsf{T}$ if all the u_is are real numbers.

Linear algebra over the complex field

One of the fundamental linear algebra ideas we've learned is how to use *linear transformations* to model input-output phenomena in which input vectors $\vec{v}$ are linearly transformed to output vectors: $\vec{w} = T(\vec{v})$. Linear transformations are vector functions of the form $T : \mathbb{R}^m \to \mathbb{R}^n$. We can *represent* these Linear transformations as an $m \times n$ matrix *with respect to* some choice of input and output bases.

These linear algebra ideas also apply to complex vectors and complex matrices. For example, a linear transformation from $T : \mathbb{C}^2 \to \mathbb{C}^2$ can be represented in terms of the matrix product

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

Each linear transformation $T : \mathbb{C}^2 \to \mathbb{C}^2$ corresponds to some 2×2 matrix $\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$ with coefficients $\alpha, \beta, \gamma, \delta \in \mathbb{C}$.

The change from real coefficients to complex coefficients has the effect of doubling the number of parameters required to describe the transformation. A 2×2 complex matrix has eight parameters, not four. Where are those eight parameters, you ask? Here:

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} \text{Re}\{\alpha\} & \text{Re}\{\beta\} \\ \text{Re}\{\gamma\} & \text{Re}\{\delta\} \end{bmatrix} + \begin{bmatrix} \text{Im}\{\alpha\} & \text{Im}\{\beta\} \\ \text{Im}\{\gamma\} & \text{Im}\{\delta\} \end{bmatrix} i.$$

Each of the four coefficients of the matrix has a real part and an imaginary part, making for a total of eight parameters to "pick" when specifying the matrix.

Similarly, to specify a vector $\vec{v} = \mathbb{C}^2$ you need to specify four parameters:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \text{Re}\{v_1\} \\ \text{Re}\{v_2\} \end{bmatrix} + \begin{bmatrix} \text{Im}\{v_1\} \\ \text{Im}\{v_2\} \end{bmatrix} i.$$

In practice, this doubling of dimensions doesn't play a role in calculations because we usually perform algebra steps with the complex coefficients and rarely split the matrices into their real and imaginary parts.

All the linear algebra techniques you've learned also work with complex numbers, as you'll see in the following examples.

Example 1: Solving systems of equations Suppose you're solving a problem that involves complex numbers and a system of two linear equations in two unknowns:

$$\begin{aligned} z_1 + 2z_2 &= 3 + i \\ 3z_1 + (9+i)z_2 &= 6 + 2i. \end{aligned}$$

You're asked to find the values of the unknowns z_1 and z_2.

The solutions z_1 and z_2 will be complex numbers, but apart from that, there's nothing special about this problem—keep in mind, linear algebra with complex numbers is the same as linear algebra with real numbers, so the techniques you learned for real numbers work just as well for complex numbers. Now let's solve this system of equations.

First observe that the system of equations can be written as a matrix-vector product:

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 9+i \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}_{\vec{z}} = \underbrace{\begin{bmatrix} 3+i \\ 6+2i \end{bmatrix}}_{\vec{b}}.$$

We've expressed the system as a 2×2 matrix A multiplying the vector of unknowns $\vec{z}$ (a 2×1 matrix) to produce a vector of constants $\vec{b}$ (another 2×1 matrix). We can solve for $\vec{z}$ by multiplying both sides of the equation by the inverse matrix A^{-1}. The inverse matrix of A is

$$A^{-1} = \begin{bmatrix} 1 + \frac{6}{3+i} & -\frac{2}{3+i} \\ -\frac{3}{3+i} & \frac{1}{3+i} \end{bmatrix}.$$

We can now compute the answer $\vec{z}$ using the equation $\vec{z} = A^{-1}\vec{b}$:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 + \frac{6}{3+i} & -\frac{2}{3+i} \\ -\frac{3}{3+i} & \frac{1}{3+i} \end{bmatrix} \begin{bmatrix} 3+i \\ 6+2i \end{bmatrix} = \begin{bmatrix} 3+i+6-4 \\ -3+2 \end{bmatrix} = \begin{bmatrix} 5+i \\ -1 \end{bmatrix}.$$

Example 2: Finding the inverse We learned several approaches for computing matrix inverses in Section 4.5. Here we'll review the procedure for computing the inverse using row operations.

Given the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 9+i \end{bmatrix},$$

first build a 2×4 array that contains A on the left side and the identity matrix $\mathbb{1}$ on the right side:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 9+i & 0 & 1 \end{array}\right].$$

We now perform the Gauss–Jordan elimination procedure on the resulting 2×4 array.

1. Subtract three times the first row from the second row ($R_2 \leftarrow R_2 - 3R_1$) to obtain
$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 3+i & -3 & 1 \end{array}\right].$$
2. Perform $R_2 \leftarrow \frac{1}{3+i}R_2$ to create a pivot in the second row:
$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & \frac{-3}{3+i} & \frac{1}{3+i} \end{array}\right].$$
3. Finally, perform $R_1 \leftarrow R_1 - 2R_2$ to obtain the RREF:
$$\left[\begin{array}{cc|cc} 1 & 0 & 1+\frac{6}{3+i} & -\frac{2}{3+i} \\ 0 & 1 & \frac{-3}{3+i} & \frac{1}{3+i} \end{array}\right].$$

The inverse of A appears on the right side of the array,
$$A^{-1} = \begin{bmatrix} 1+\frac{6}{3+i} & -\frac{2}{3+i} \\ -\frac{3}{3+i} & \frac{1}{3+i} \end{bmatrix}.$$

Example 3: Linear transformations as matrices The effect of multiplying a vector $\vec{v} \in \mathbb{C}^n$ by a matrix $M \in \mathbb{C}^{m \times n}$ is the same as applying a linear transformation $T_M : \mathbb{C}^n \to \mathbb{C}^m$ to the vector:
$$\vec{w} = M\vec{v} \qquad \Leftrightarrow \qquad \vec{w} = T_M(\vec{v}).$$
The opposite is also true—any linear transformation T can be *represented* as a matrix product with some matrix M_T:
$$\vec{w} = T(\vec{v}) \qquad \Leftrightarrow \qquad \vec{w} = M_T\vec{v}.$$
We'll use a simple example to review the procedure for finding the matrix representation of a linear transformation.

Consider the linear transformation $T : \mathbb{C}^2 \to \mathbb{C}^2$, which produces the input-output pairs
$$T\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}3\\2i\end{bmatrix} \qquad \text{and} \qquad T\left(\begin{bmatrix}0\\2\end{bmatrix}\right) = \begin{bmatrix}2\\4+4i\end{bmatrix}.$$
How can we use the information provided above to find the matrix representation of the linear transformation T?

To obtain the matrix representation of T with respect to a given basis, we need to combine, as columns, the outputs of T for the n elements of that basis:
$$M_T = \begin{bmatrix} | & | & & | \\ T(\vec{e}_1) & T(\vec{e}_2) & \cdots & T(\vec{e}_n) \\ | & | & & | \end{bmatrix},$$

where the set $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ is a basis for the input space.

The problem statement gives us the information needed for the first column of M_T, but we're not given the output of T for $\hat{e}_2$. However, we can work around this limitation since we know T is *linear*. The property $T(\alpha\vec{v}) = \alpha T(\vec{v})$ implies

$$T\left(2\begin{bmatrix}0\\1\end{bmatrix}\right) = 2\begin{bmatrix}1\\2+2i\end{bmatrix} \qquad \Rightarrow \qquad T\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}1\\2+2i\end{bmatrix}.$$

Combining the information for $T(\hat{e}_1)$ and $T(\hat{e}_2)$, we obtain the matrix representation of T:

$$M_T = \begin{bmatrix}3 & 1\\2i & 2+2i\end{bmatrix}.$$

Complex eigenvalues

The main reason why I want you to learn about linear algebra with complex vectors is so we can complete the important task of classifying the basic types of linear transformations in terms of their eigenvalues. Recall that projections obey $\Pi = \Pi^2$ and have eigenvalues zero or one, and reflections have at least one eigenvalue equal to -1.

What are the eigenvalues of rotation matrices? The eigenvalues of a matrix A are the roots of its characteristic polynomial $p_A(\lambda) = \det(A - \lambda\mathbb{1})$. To find the eigenvalues of the rotation matrix R_θ we defined in Section 6.2 (page 241), we must find the solutions of the equation $p_{R_\theta}(\lambda) = 0$:

$$\begin{aligned}0 &= p_{R_\theta}(\lambda)\\ &= \det(R_\theta - \lambda\mathbb{1})\\ &= \det\left(\begin{bmatrix}\cos\theta - \lambda & -\sin\theta\\ \sin\theta & \cos\theta - \lambda\end{bmatrix}\right)\\ &= (\cos\theta - \lambda)^2 + \sin^2\theta.\end{aligned}$$

To solve for λ, first move $\sin^2\theta$ to the other side of the equation

$$-\sin^2\theta = (\cos\theta - \lambda)^2,$$

then take the square root on both sides:

$$\cos\theta - \lambda = \pm\sqrt{-\sin^2\theta} = \pm\sqrt{-1}\sin\theta = \pm i\sin\theta.$$

The eigenvalues of R_θ are $\lambda_1 = \cos\theta + i\sin\theta$ and $\lambda_2 = \cos\theta - i\sin\theta$. Using Euler's formula (see page 106) we can express the eigenvalues more compactly as $\lambda_1 = e^{i\theta}$ and $\lambda_2 = e^{-i\theta}$. What's interesting here is that complex numbers emerge as answers to a matrix problem that was originally stated in terms of real variables.

This is not a coincidence: complex exponentials are in many ways the natural way to talk about rotations, periodic motion, and waves. If you pursue a career in math, physics, or engineering you'll use complex numbers and Euler's equation on a daily basis.

Special types of matrices

We'll now define a few special types of matrices with complex coefficients. These matrices are analogous to the special matrices we defined in Section 7.2, but their definitions are adapted to use the Hermitian conjugate operation †.

Unitary matrices

A matrix U is *unitary* if it obeys $U^\dagger U = \mathbb{1}$. The norm of the determinant of a unitary matrix is 1, $|\det(U)| = 1$. For an $n \times n$ matrix U, the following statements are equivalent:

- U is unitary.
- The columns of U form an orthonormal basis.
- The rows of U form an orthonormal basis.
- The inverse of U is $U^\dagger$.

Unitary matrices are the complex analogues of orthogonal matrices. Indeed, if a unitary matrix U has real coefficients, then $U^\dagger = U^\mathsf{T}$ and we have $U^\mathsf{T}U = \mathbb{1}$, which is the definition of an orthogonal matrix.

Hermitian matrices

A Hermitian matrix H is equal to its own Hermitian transpose:

$$H^\dagger = H \qquad \Leftrightarrow \qquad h_{ij} = \overline{h_{ji}}, \quad \text{for all } i, j.$$

Hermitian matrices are complex-number analogues of symmetric matrices.

A Hermitian matrix H can be freely moved from one side to the other in a complex inner product:

$$\langle H\vec{x}, \vec{y}\rangle = (H\vec{x})^\dagger \vec{y} = \vec{x}^\dagger H^\dagger \vec{y} = \vec{x}^\dagger\, (H\vec{y}) = \langle \vec{x}, H\vec{y}\rangle.$$

The eigenvalues of Hermitian matrices are real numbers.

Normal matrices

Previously, we defined the set of real normal matrices to be matrices that satisfy $A^\mathsf{T}A = AA^\mathsf{T}$. For matrices with complex coefficients, the definition of a normal matrix uses the dagger: $A^\dagger A = AA^\dagger$.

Consulting the concept map in Figure 7.1 on page 276 will help you see the parallels between the different types of special matrices. I realize there's a lot of new terminology to absorb all at once, so don't worry about remembering everything. The main idea is to know these special types of matrices exist—not to know *everything* about them.

Inner product for complex vectors

The complex inner product is an operation of the form:

$$\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}.$$

The inner product $\langle \vec{u}, \vec{v} \rangle$ for real vectors is equivalent to the matrix product between the row vector $\vec{u}^T$ and the column vector $\vec{v}$. Extending the notion of inner product to work with complex vectors requires a modification to the inner product formula. The inner product for vectors $\vec{u}, \vec{v} \in \mathbb{C}^n$ is defined as

$$\langle \vec{u}, \vec{v} \rangle \equiv \sum_{i=1}^{n} \overline{u_i} v_i \equiv \vec{u}^\dagger \vec{v}.$$

The formula is similar to the inner product formula for real vectors, but uses the Hermitian transpose † instead of the regular transpose ^{T}. The inner product of two vectors $\vec{u}, \vec{v} \in \mathbb{C}^3$ is

$$\langle \vec{u}, \vec{v} \rangle = \bar{u}_1 v_1 + \bar{u}_2 v_2 + \bar{u}_3 v_3 = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \bar{u}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \vec{u}^\dagger \vec{v}.$$

This dagger thing is very important. Using the definition of the inner product with a dagger on the first entry ensures the complex inner product will obey the positive semidefinite criterion (see page 281). The inner product of a vector $\vec{v} \in \mathbb{C}^3$ with itself is

$$\langle \vec{v}, \vec{v} \rangle \equiv \vec{v}^\dagger \vec{v} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 & \bar{v}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = |v_1|^2 + |v_2|^2 + |v_3|^2,$$

where $|v_i|^2 = \bar{v}_i v_i$ is the magnitude-squared of the coefficient $v_i \in \mathbb{C}$. The magnitudes of the complex coefficients are nonnegative real numbers, so the sum of their squares is also a nonnegative real number. Therefore, the complex inner product satisfies the positive semidefinite requirement $\langle \vec{v}, \vec{v} \rangle \geq 0$ for inner products.

Length of a complex vector

The complex inner product induces the following complex norm:

$$\|\vec{v}\| \equiv \sqrt{\langle \vec{v}, \vec{v} \rangle} = \sqrt{\vec{v}^\dagger \vec{v}} = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}\,.$$

The norm for complex vectors satisfies the positive semidefinite requirement $\|\vec{v}\| \geq 0$ for norms (see page 283).

Example Calculate the norm of the vector $\vec{v} = (2+i, 3, 5i)$.

The Hermitian transpose of the row vector $\vec{v}$ is the column vector $\vec{v}^\dagger = (2-i, 3, -5i)^\mathsf{T}$. The norm of $\vec{v}$ is equal to the square root of $\langle \vec{v}, \vec{v} \rangle$ so $\|\vec{v}\| = \sqrt{\langle \vec{v}, \vec{v} \rangle} = \sqrt{(2-i)(2+i) + 3^2 + (-5i)(5i)} = \sqrt{4+1+9+25} = \sqrt{39}$.

Complex inner product spaces

A real inner product space is an abstract vector space $(V, \mathbb{R}, +, \cdot)$ for which we've defined an inner product operation $\langle \mathbf{u}, \mathbf{v} \rangle$ which obeys (1) the symmetric property, (2) the linearity property, and (3) the positive semidefinite property.

Similarly, a complex inner product space is an abstract vector space $(V, \mathbb{C}, +, \cdot)$ with an inner product operation $\langle \mathbf{u}, \mathbf{v} \rangle$ that satisfies the following criteria for all $\mathbf{u}, \mathbf{v}, \mathbf{v}_1, \mathbf{v}_2 \in V$ and $\alpha, \beta \in \mathbb{C}$:

- Conjugate symmetric: $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$
- Linear: $\langle \mathbf{u}, \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 \rangle = \alpha\langle \mathbf{u}, \mathbf{v}_1 \rangle + \beta\langle \mathbf{u}, \mathbf{v}_2 \rangle$
- Positive semidefinite: $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ for all $\mathbf{u} \in V$ with $\langle \mathbf{u}, \mathbf{u} \rangle = 0$ if and only if $\mathbf{u} = \mathbf{0}$

The conjugate symmetry property $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$ ensures the inner product of a vector with itself is a real number: $\langle \mathbf{u}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{u} \rangle} \in \mathbb{R}$.

Example The Hilbert–Schmidt inner product for matrices $A, B \in \mathbb{C}^{m \times n}$ is defined as

$$\langle A, B \rangle_{\text{HS}} \equiv \text{Tr}\big(A^\dagger B\big) = \sum_{i=1}^{n} \langle A\vec{e}_i, B\vec{e}_i \rangle.$$

The product $A\vec{e}_i$ has the effect of "selecting" the i^{th} column of the matrix A; we can consider the Hilbert–Schmidt inner product of matrices A and B as the sum of the vector inner products of the columns of the two matrices.

We can also define the Hilbert–Schmidt norm for matrices:

$$||A||_{\text{HS}} \equiv \sqrt{\langle A, A \rangle} = \sqrt{\text{Tr}(A^\dagger A)} = \left[\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right]^{\frac{1}{2}}.$$

The Hilbert–Schmidt norm is the square root of the sum of the squared-magnitudes of the entries of the matrix.

The Hilbert–Schmidt inner product and norm are sometimes called *Frobenius inner product* and *Frobenius norm*, respectively.

Singular value decomposition

The singular value decomposition we introduced for real matrices in Section 7.6 also applies to matrices with complex entries.

The singular value decomposition of a matrix $A \in \mathbb{C}^{m\times n}$ is a way to express A as the product of three matrices:

$$A = U\Sigma V^\dagger.$$

The $m \times m$ unitary matrix U consists of *left singular vectors* of A. The $m \times n$ matrix Σ contains the *singular values* σ_i on the diagonal. The $n \times n$ unitary matrix $V^\dagger$ consists of *right singular vectors*.

The singular values σ_i of A are the positive square roots of the eigenvalues of the matrix $AA^\dagger$. To find the matrix of left singular vectors U, calculate the eigenvectors of $AA^\dagger$ and pack them as columns. To find the matrix of right singular vectors $V^\dagger$, calculate the eigenvectors $A^\dagger A$, pack them as columns in a matrix V, then take the Hermitian transpose of this matrix.

The Hilbert–Schmidt norm of a matrix $A \in \mathbb{C}^{m\times n}$ is equal to the square root of the sum of the squares of its singular values:

$$||A||_{\text{HS}} = \sqrt{\text{Tr}(A^\dagger A)} = \sqrt{\sum_{i=1}^{n} \sigma_i^2}.$$

This important fact about matrices shows a connection between the "size" of a matrix and the size of its singular values. Each singular value σ_i corresponds to the strength of the effect of A on the i^{th} subspaces spanned by its left and right singular vectors.

The singular value decomposition is used in many algorithms and procedures to uncover the inner structure of matrices. The machine learning technique called *principal component analysis* (PCA) corresponds to applying the SVD to a data matrix. Alternatively, you can think of the PCA as applying an eigendecomposition of the *covariance matrix* of the data.

Explanations

Complex eigenvectors

The characteristic polynomial of the rotation matrix R_θ is $p(\lambda) = (\cos\theta - \lambda)^2 + \sin^2\theta = 0$. The eigenvalues of R_θ are $\lambda_1 = \cos\theta + i\sin\theta = e^{i\theta}$ and $\lambda_2 = \cos\theta - i\sin\theta = e^{-i\theta}$. What are its eigenvectors?

Before we get into the eigenvector calculation, I want to show you a useful trick for rewriting cos and sin expressions in terms of complex

exponential functions. Recall Euler's equation, $e^{i\theta} = \cos\theta + i\sin\theta$. Using this equation and the analogous expression for $e^{-i\theta}$, we can obtain the following expressions for $\cos\theta$ and $\sin\theta$:

$$\cos\theta = \frac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right), \qquad \sin\theta = \frac{1}{2i}\left(e^{i\theta} - e^{-i\theta}\right).$$

Try calculating the right-hand side in each case to verify the accuracy of each expression. These formulas are useful because they allow us to rewrite expressions of the form $e^{i\theta}\cos\phi$ as $e^{i\theta}\frac{1}{2}\left(e^{i\phi} + e^{-i\phi}\right) = \frac{1}{2}\left(e^{i(\theta+\phi)} + e^{i(\theta-\phi)}\right)$, which is simpler.

Let's now see how to find the eigenvector $\vec{e}_{\lambda_1} \equiv (\alpha, \beta)^\mathsf{T}$ associated with the eigenvalue $\lambda_1 = e^{i\theta}$. The eigenvalue equation for the eigenvalue $\lambda_1 = e^{i\theta}$ is

$$R_\theta \vec{e}_{\lambda_1} = e^{i\theta}\vec{e}_{\lambda_1} \qquad \Leftrightarrow \qquad \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = e^{i\theta}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}.$$

We're looking for the coefficients α and β.

Do you remember how to find eigenvectors? Don't worry if you've forgotten—this is why we have this review chapter! We'll go through the problem in detail. Brace yourself though, because the calculation is quite long.

The "finding the eigenvector(s) of A for the eigenvalue λ_1" task is carried out by finding the *null space* of the matrix $(A - \lambda_1 \mathbb{1})$. We rewrite the eigenvalue equation stated above as

$$(R_\theta - e^{i\theta}\mathbb{1})\vec{e}_{\lambda_1} = 0 \quad \Leftrightarrow \quad \begin{bmatrix} \cos\theta - e^{i\theta} & -\sin\theta \\ \sin\theta & \cos\theta - e^{i\theta} \end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix}.$$

It's now clear that the finding-the-eigenvectors procedure corresponds to a null space calculation.

Let's use the cos-rewriting trick to simplify $\cos\theta - e^{i\theta}$:

$$\begin{aligned}
\cos\theta - e^{i\theta} &= \tfrac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right) - e^{i\theta} \\
&= \tfrac{1}{2}e^{i\theta} + \tfrac{1}{2}e^{-i\theta} - e^{i\theta} = \tfrac{-1}{2}e^{i\theta} + \tfrac{1}{2}e^{-i\theta} = \tfrac{-1}{2}\left(e^{i\theta} - e^{-i\theta}\right) \\
&= -i\tfrac{1}{2i}\left(e^{i\theta} - e^{-i\theta}\right) \\
&= -i\sin\theta.
\end{aligned}$$

We substitute this simplified expression in the two places where it appears, and do some row operations to simplify the matrix:

$$\begin{bmatrix} -i\sin\theta & -\sin\theta \\ \sin\theta & -i\sin\theta \end{bmatrix} \sim \begin{bmatrix} \sin\theta & -i\sin\theta \\ \sin\theta & -i\sin\theta \end{bmatrix} \sim \begin{bmatrix} \sin\theta & -i\sin\theta \\ 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 1 & -i \\ 0 & 0 \end{bmatrix}.$$

We can now solve the null space problem. Observe that the second column of the matrix does not contain a pivot, so β is a free variable,

which we'll denote $s \in \mathbb{R}$. We thus obtain the equations:

$$\begin{bmatrix} 1 & -i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} 1\alpha + (-is) &= 0, \\ 0 &= 0. \end{aligned}$$

Solving for α in terms of s, we find $\alpha = is$, and therefore the solution is $(\alpha, \beta)^\mathsf{T} = (is, s)$. The eigenspace that corresponds to the eigenvalue $\lambda_1 = e^{i\theta}$ is the null space of the matrix $(R_\theta - e^{i\theta}\mathbb{1})$:

$$\mathcal{N}(R_\theta - e^{i\theta}\mathbb{1}) = \left\{ \begin{bmatrix} is \\ s \end{bmatrix}, \ \forall s \in \mathbb{R} \right\} = \text{span}\left\{ \begin{bmatrix} i \\ 1 \end{bmatrix} \right\}.$$

After all this work, we've finally obtained an eigenvector $\vec{e}_{\lambda_1} = (i, 1)^\mathsf{T}$ that corresponds to the eigenvalue $\lambda_1 = e^{i\theta}$. Let's verify that the vector we obtained satisfies the eigenvalue equation $R_\theta \vec{e}_{\lambda_1} = e^{i\theta}\vec{e}_{\lambda_1}$:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ 1 \end{bmatrix} = \begin{bmatrix} i\cos\theta - \sin\theta \\ i\sin\theta + \cos\theta \end{bmatrix} = \begin{bmatrix} i(\cos\theta + i\sin\theta) \\ \cos\theta + i\sin\theta \end{bmatrix} = e^{i\theta} \begin{bmatrix} i \\ 1 \end{bmatrix}.$$

The eigenvector for the eigenvalue $\lambda_2 = e^{-i\theta}$ is $\vec{e}_{\lambda_2} = (i, -1)^\mathsf{T}$. Verify that it satisfies the eigenvalue equation $R_\theta \vec{e}_{\lambda_2} = e^{-i\theta}\vec{e}_{\lambda_2}$.

I know it was quite a struggle to find the eigenvectors of this rotation matrix, but this the case in general when finding eigenvectors. You must complete the null space calculation steps for each eigenspace, and this takes a long time. Be sure to practice finding eigenvectors by hand—I can pretty much guarantee you'll need this skill on your linear algebra final. And don't forget to give yourself a pat on the back when you're done!

Properties of the Hermitian transpose operation

The Hermitian transpose obeys the following properties:

- $(A + B)^\dagger = A^\dagger + B^\dagger$
- $(AB)^\dagger = B^\dagger A^\dagger$
- $(ABC)^\dagger = C^\dagger B^\dagger A^\dagger$
- $(A^\dagger)^{-1} = (A^{-1})^\dagger$

Note these are the same properties as the regular transpose operation from Section 3.3 (see page 123).

Conjugate linearity in the first input

The complex inner product we defined is linear in the second entry and *conjugate-linear* in the first entry:

$$\langle \vec{v}, \alpha\vec{a} + \beta\vec{b} \rangle = \alpha\langle \vec{v}, \vec{a} \rangle + \beta\langle \vec{v}, \vec{b} \rangle,$$
$$\langle \alpha\vec{a} + \beta\vec{b}, \vec{w} \rangle = \overline{\alpha}\langle \vec{a}, \vec{w} \rangle + \overline{\beta}\langle \vec{b}, \vec{w} \rangle.$$

Keep this in mind every time you deal with complex inner products. The complex inner product is not symmetric since it requires that the complex conjugation be performed on the first input. Remember, instead of $\langle \vec{v}, \vec{w} \rangle \neq \langle \vec{w}, \vec{v} \rangle$, we have $\langle \vec{v}, \vec{w} \rangle = \overline{\langle \vec{w}, \vec{v} \rangle}$.

The choice of complex conjugation in the first entry is a matter of convention. In this text, we *defined* the inner product $\langle \cdot, \cdot \rangle$ with the † operation on the first entry, which is known as the *physics convention*. Some old mathematics texts define the inner product of complex vectors using the complex conjugation on the second entry, which makes the inner product linear in the first entry and conjugate-linear in the second entry. This convention is fine, too. The choice of convention doesn't matter, as long as one of the entries is conjugated to ensure the inner product obeys the positive semidefinite requirement $\langle \vec{u}, \vec{u} \rangle \geq 0$.

Function inner product

In the section on inner product spaces, we discussed the notion of the vector space of all real-valued functions of a real variable $f : \mathbb{R} \to \mathbb{R}$, and defined an inner product between functions:

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x) g(x) \, dx.$$

Suppose we have two complex-valued functions $f(x)$ and $g(x)$:

$$f \colon \mathbb{R} \to \mathbb{C}, \qquad g \colon \mathbb{R} \to \mathbb{C}.$$

We define the inner product for complex-valued functions as

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \overline{f(x)} g(x) \, dx.$$

The complex conjugation of one of the functions ensures that the inner product of a function with itself results in a real number. The function inner product measures the *overlap* between $f(x)$ and $g(x)$.

Linear algebra over other fields

We can carry out linear algebra calculations over any *field*. A field is a set of numbers for which an addition, subtraction, multiplication, and division operation are defined. The addition and multiplication operations we define must be associative and commutative, and multiplication must distribute over addition. Furthermore, a field must contain an additive identity element (denoted 0) and a multiplicative identity element (denoted 1). The properties of a field are essentially all the properties of the numbers you're familiar with: $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$.

The focus of our discussion in this section was to show the linear algebra techniques we learned for manipulating real numbers work equally well with the complex numbers. This shouldn't be too surprising since, after all, linear algebra manipulations boil down to arithmetic manipulations of the coefficients of vectors and matrices. As both real numbers and complex numbers can be added, subtracted, multiplied, and divided, we can study linear algebra over both $\mathbb{R}$ and $\mathbb{C}$.

We can also perform linear algebra over *finite fields*. A *finite field* is a set $\mathbb{F}_q \equiv \{0, 1, 2, \ldots, q-1\}$, where q is a prime number or the power of a prime number. All the arithmetic operations in this field are performed *modulo* the number q, which means all arithmetic operations must result in answers in the set $\mathbb{F}_q \equiv \{0, 1, 2, \ldots, q-1\}$. If the result of an operation falls outside this set, we either add or subtract q until the number falls in the set $\mathbb{F}_q$. Consider the finite field $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$. To add two numbers in $\mathbb{F}_5$, proceed as follows:

$$\begin{aligned}(3+3) \bmod 5 &= 6 \mod 5 \qquad \text{(too big, so subtract 5)} \\ &= 1 \mod 5.\end{aligned}$$

Similarly, for subtraction,

$$\begin{aligned}(1-4) \bmod 5 &= (-3) \bmod 5 \qquad \text{(too small, so add 5)} \\ &= 2 \bmod 5.\end{aligned}$$

The field of binary numbers $\mathbb{F}_2 \equiv \{0, 1\}$ is an important finite field used in many areas of communications, engineering, and cryptography. In the next chapter we'll discuss the one-time crypto system which allows for secure communication of messages encoded in binary (Section 8.9). We'll also discuss the error-correcting codes used that enable the reliable transmission of information over noisy communication channels (Section 8.10). For example, the data packets that your cell phone sends over the radio waves are first linearly encoded using a matrix-vector product operation carried out over the field $\mathbb{F}_2$.

At first hand, thinking of linear algebra over the finite field $\mathbb{F}_q \equiv \{0, 1, 2, \ldots, q-1\}$ may seem complicated, but don't worry about it. It's the same stuff—you just have to mod$q$ every arithmetic calculation. All you your intuition about dimensions and orthogonality, and all the computational procedures you know are still applicable.

The field of rational numbers $\mathbb{Q}$ is another example of a field that's often used in practice. Solving systems of equations using rational numbers on computers is interesting because the answers obtained are exact—using rational numbers allows us to avoid many of the numerical accuracy problems associated with floating point numbers.

Discussion

The adjoint operator

Though we used the term *Hermitian transpose* and the notation $A^\dagger$ throughout this section, it's worth commenting that mathematicians prefer the term *adjoint* for the same operation, and denote it A^*. Recall we previously discussed the concept of an *adjoint linear transformation* $T_{M^\intercal}:\mathbb{R}^m \to \mathbb{R}^n$, which corresponds to the multiplication of a matrix M by a row vector from the left $T_{M^\intercal}(\vec{a}) \equiv \vec{a}^\intercal M$ (see page 205). We didn't use the term "transpose" then because transposing is something you do to matrices. Instead, we used the math term *adjoint*, which precisely describes the notion of the "transpose of a linear transformation." Since we're on the topic of math terminology, it should be noted that some mathematicians use the term *adjoint operator* instead of *adjoint linear transformation*, since they call *operators* what we call *linear transformations*.

Matrix quantum mechanics

Guess what? Understanding linear algebra over the complex field means you understand quantum mechanics! Quantum mechanics unfolds in a complex inner product space (called a Hilbert space). If you understood the material in this section, you should be able to understand the axioms of quantum mechanics at no additional mental cost. If you're interested in this kind of stuff you should read Chapter 10.

Exercises

E7.18 Calculate (a) $(2+5i)-(3+4i)$, (b) $(2+5i)(3+4i)$, and (c) $(2+5i)/(3+4i)$.

7.8 Theory problems

It's now time to test your understanding by solving some problems.

P7.1 Yuna wants to cheat on her exam, she needs your help. Please help her to compute eigenvalues for the following matrices and slip her the piece of paper carefully so the teacher doesn't notice. Yuna will give you a chocolate bar to thank you.

a) $\begin{bmatrix} 3 & 1 \\ 12 & 2 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 2 \\ 0 & 1 & 0 \end{bmatrix}$

P7.2 Compute the eigenvalues of the matrix $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$.

P7.3 Show that the vector $\vec{e}_1 = (1, \frac{1}{\varphi})^\mathsf{T}$ and $\vec{e}_2 = (1, -\varphi)^\mathsf{T}$ are eigenvectors of the matrix $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. What are the eigenvalues associated with these eigenvectors?

Hint: Compute $A\vec{e}_1$ and see what happens. Use the fact that φ satisfies the equation $\varphi^2 - \varphi - 1 = 0$ to simplify expressions.

P7.4 We can write the matrix $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ as the product of three matrices $Q\Lambda X$, where Q contains the eigenvectors of A, and Λ contains its eigenvalues:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ \frac{1}{\varphi} & -\varphi \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \varphi & 0 \\ 0 & \frac{-1}{\varphi} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}}_{X}.$$

Find the matrix X.

P7.5 Find eigenvalues for the matrices below:

a) $\begin{bmatrix} 4 & 2 \\ 0 & 5 \end{bmatrix}$ **b)** $\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$ **c)** $\begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 4 & -1 \end{bmatrix}$ **d)** $\begin{bmatrix} -3 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 1 & -1 \end{bmatrix}$

P7.6 Compute the eigenvalues and eigenvectors of these matrices:

a) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -1 & 4 \end{bmatrix}$

P7.7 Given $A = \begin{bmatrix} 2 & 2 \\ 5 & -1 \end{bmatrix}$, find A^{10}.

P7.8 Consider the sequence of triples $\{(x_n, y_n, z_n)\}_{n=0,1,2,\ldots}$ produced according to the formula:

$$\underbrace{\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{M} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix}$$

Give a formula for $(x_\infty, y_\infty, z_\infty)$ in terms of (x_0, y_0, z_0). See `youtu.be/mX0NB9IyYpU` to see how this recurrence relation is related to "surface smoothing" algorithms used in 3D graphics.

Hint: Compute the eigenvalues λ_1, λ_2, and λ_3 of the matrix M. What will become of the eigenvalues if you raise them to the power ∞?

P7.9 Check if the following matrices are orthogonal or not:

a) $\begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix}$ **c)** $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

P7.10 Let V be the set of two dimensional vectors of real numbers, with addition defined as $(a_1, a_2) + (b_1, b_2) = (a_1 + b_1, a_2 b_2)$ and scalar multiplication defined as $c \cdot (a_1, a_2) = (ca_1, a_2)$. Is $(V, \mathbb{R}, +, \cdot)$ a vector space? Justify your answer.

P7.11 Let $V = \{(a_1, a_2)\}$, with $a_1, a_2 \in \mathbb{R}$. Define vector addition as $(a_1, a_2) + (b_1, b_2) = (a_1 + 2b_1, a_2 + 3b_2)$ and scalar multiplication as $c \cdot (a_1, a_2) = (ca_1, ca_2)$. Is $(V, \mathbb{R}, +, \cdot)$ a vector space? Justify your answer.

P7.12 Prove the Cauchy–Schwarz inequality $|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\| \|\mathbf{v}\|$.

Hint: It is true that $\|\mathbf{a}\| > 0$ for any vector $\mathbf{a}$. Use this fact to expand the expression $\|\mathbf{u} - c\mathbf{v}\| > 0$, choosing $c = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle}$.

P7.13 Prove the triangle inequality $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$.

Hint: Compute $\|\mathbf{u} + \mathbf{v}\|$ as an inner product and simplify the expression using the fact $\langle \mathbf{a}, \mathbf{b} \rangle \leq \|\mathbf{a}\| \|\mathbf{b}\|$ for all vectors $\mathbf{a}$ and $\mathbf{b}$.

P7.14 Perform the Gram–Shmidt orthogonalization procedure on the following basis for $\mathbb{R}^2$: $\{(0, 1), (-1, 0)\}$.

P7.15 Perform Gram–Shmidt orthogonalization on vectors $\vec{v}_1 = (1, 1)$ and $\vec{v}_2 = (0, 1)$ to obtain an orthonormal basis.

P7.16 Convert the vectors $(3, 1)$ and $(-1, 1)$ into an orthonormal basis.

P7.17 Find the eigendecomposition of the following matrix:

$$A = \begin{bmatrix} 2 & 0 & -5 \\ 0 & 2 & 0 \\ 0 & 0 & -3 \end{bmatrix}.$$

P7.18 Compute the following expressions: a) $|3i - 4|$; b) $\overline{2 - 3i}$; c) $(3i - 1) + \overline{3 - 2i}$; d) $|\overline{-3i} - \overline{-4i + 5}|$

P7.19 Given matrices A, B, and C below, find $A + B$, CB and $(2 + i)B$.

$$A = \begin{bmatrix} 2 + i & -1 + 2i \\ 3 + 2i & -2i \end{bmatrix} \quad B = \begin{bmatrix} 2 - i & 3 - 2i \\ 5 + i & -5 + 5i \end{bmatrix} \quad C = \begin{bmatrix} 1 + 2i & i \\ 3 - i & 8 \\ 4 + 2i & 1 - i \end{bmatrix}$$

P7.20 Find the eigenvalues of the following matrices:

a) $\begin{bmatrix} 3 & -2 \\ 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 3 & -9 \\ 4 & -3 \end{bmatrix}$ **c)** $\begin{bmatrix} 3 & -13 \\ 5 & 1 \end{bmatrix}$

P7.21 Give a basis for the vector space of 3×3 diagonal matrices.

P7.22 What is the dimension of the vector space of 3×3 symmetric matrices.

Hint: See page 272 for definition.

P7.23 How many elements are there in a basis for the vector space of 3×3 Hermitian matrices.

Hint: See page 305 for definition.

P7.24 A matrix is *nilpotent* if it becomes the zero matrix when repeatedly multiplied by itself. We say A is nilpotent if $A^k = 0$ for some power k. A nilpotent matrix has only the eigenvalue zero, and hence its trace and determinant are zero. Are the following matrices nilpotent?

a) $\begin{bmatrix} -2 & 4 \\ -1 & 2 \end{bmatrix}$ **b)** $\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$ **c)** $\begin{bmatrix} -3 & 2 & 1 \\ -3 & 2 & 1 \\ -3 & 2 & 1 \end{bmatrix}$ **d)** $\begin{bmatrix} 1 & 1 & 4 \\ 3 & 0 & -1 \\ 5 & 2 & 7 \end{bmatrix}$

e) $\begin{bmatrix} 45 & -22 & -19 \\ 33 & -16 & -14 \\ 69 & -34 & -29 \end{bmatrix}$ **f)** $\begin{bmatrix} 5 & -3 & 2 \\ 15 & -9 & 6 \\ 10 & -6 & 4 \end{bmatrix}$

P7.25 Determine all the eigenvalues of $A = \left[\begin{smallmatrix} 1+i & 1 \\ 2 & 1-i \end{smallmatrix}\right]$. For each eigenvalue λ of A, find the set of eigenvectors corresponding to λ. Determine whether or not A is diagonalizable and if so find an invertible matrix Q and a diagonal matrix Λ such that $Q^{-1}AQ = \Lambda$.

P7.26 Let $\vec{v}_1, \vec{v}_2, \vec{v}_3$ be vectors in the inner product space V. Given $\langle \vec{v}_1, \vec{v}_2 \rangle = 3$, $\langle \vec{v}_2, \vec{v}_3 \rangle = 2$, $\langle \vec{v}_1, \vec{v}_3 \rangle = 1$, $\langle \vec{v}_1, \vec{v}_1 \rangle = 1$, and $\langle \vec{v}_2, \vec{v}_1 + \vec{v}_2 \rangle = 13$, calculate:

a) $\langle \vec{v}_1, 2\vec{v}_2 + 3\vec{v}_3 \rangle$ **b)** $\langle 2\vec{v}_1 - \vec{v}_2, \vec{v}_1 + \vec{v}_3 \rangle$ **c)** $\|\vec{v}_2\|$

P7.27 Consider the basis $B_a = \{1 + ix, 1 + x + ix^2, 1 + 2ix\}$.

(a) Show that B_a is a basis for the space of polynomials with complex coefficients of degree at most 2.

Chapter 8

Applications

In this chapter we'll learn about applications of linear algebra. We'll cover a wide range of topics from different areas of science, business, and technology to give you an idea of the spectrum of things you can do using matrix and vector algebra. Don't worry if you're not able to follow all the details in each section—we're taking a shotgun approach here, covering topics from many different areas in the hope to hit some that will be of interest to you. Note that most of the material covered in this chapter is not likely to show up on your linear algebra final, so no pressure on you, this is just for fun.

Before we start, I want to say a few words about scientific ethics. Linear algebra is a powerful tool for solving problems and modelling the real world. But with great power comes great responsibility. I hope you'll make an effort to think about the ethical implications when you use linear algebra to solve problems. Certain applications of linear algebra, like building weapons, interfering with crops, and building mathematically-complicated financial scams are clearly evil, so you should avoid them. Other areas where linear algebra can be applied are not so clear cut: perhaps you're building a satellite localization service to find missing people in emergency situations, but the same technology might end up being used by governments to spy on and persecute your fellow citizens. Do you want to be the person responsible for bringing about an Orwellian state? All I ask of you is to make a quick "System check" before you set to work on a project. Don't just say "It's my job" and go right ahead. If you find what you're doing at work to be unethical then maybe you should find a different job. There are a lot of jobs out there for people who know math, and if the bad guys can't hire quality people like you, their power will decrease—and that's a good thing.

Onto the applications.

8.1 Balancing chemical equations

Suppose you're given the chemical equation $H_2 + O_2 \to H_2O$, which indicates that hydrogen molecules (H_2) and oxygen molecules (O_2) can combine to produce water molecules (H_2O). Chemical equations describe how a set of *reactants* are transformed to a set of *products*. In this case the reactants are hydrogen and oxygen molecules and the products are water molecules.

The equation $H_2 + O_2 \to H_2O$ is misleading since it doesn't tell us the correct *stoichiometric ratios*: how much of each type of molecule is consumed and produced. We say the equation is not *balanced*. To *balance* the equation we must add coefficients in front of each reactant and each product so the total number of atoms on both sides of the reaction is the same: $2H_2 + O_2 \to 2H_2O$. Two hydrogen molecules are required for each oxygen molecule, since water molecules contain one oxygen and two hydrogen atoms.

Let's look at another example. The combustion of methane gas is described by the following chemical equation:

$$CH_4 + O_2 \;\to\; CO_2 + H_2O\,.$$

We want to answer the following two questions. How many molecules of oxygen will be consumed during the combustion of 1000 molecules of methane? How many CO_2 molecules will be produced as a result?

Before we can answer such questions, we must find the coefficients a, b, c, and d that balance the methane-combustion equation:

$$aCH_4 + bO_2 \;\to\; cCO_2 + dH_2O.$$

For the equation to be balanced, the same number of atoms of each type must appear on each side of the equation. For the methane combustion reaction to be balanced, the following equations must be satisfied:

$$\begin{aligned} a &= c && \text{for } C \text{ atoms to be balanced,}\\ 4a &= 2d && \text{for } H \text{ atoms to be balanced,}\\ 2b &= 2c + d && \text{for } O \text{ atoms to be balanced.} \end{aligned}$$

We can move the c and d terms to the left side of each equation and rewrite the system of equations as a matrix equation:

$$\begin{aligned} a - c &= 0\\ 4a - 2d &= 0\\ 2b - 2c - d &= 0 \end{aligned} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} 1 & 0 & -1 & 0\\ 4 & 0 & 0 & -2\\ 0 & 2 & -2 & -1 \end{bmatrix}}_{A} \begin{bmatrix} a\\ b\\ c\\ d \end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix}.$$

We're looking for the vector of coefficients $\vec{x} = (a, b, c, d)$ which is the solution to the null space problem $A\vec{x} = \vec{0}$. See **E5.7** for details. The RREF of A contains three pivots, one free variable, and the solution to the null space problem is

$$\{(a, b, c, d)\} = \text{span}\{(\tfrac{1}{2}, 1, \tfrac{1}{2}, 1)\}.$$

The solution is a one-dimensional infinite space spanned by the vector $(\frac{1}{2}, 1, \frac{1}{2}, 1)$. The solution space is infinite since any balanced equation will remain balanced if we double, or triple the amount of reactants and products. Choosing the coefficients as suggested by the solution to the null space problem gives $\frac{1}{2}\text{CH}_4 + \text{O}_2 \to \frac{1}{2}\text{CO}_2 + \text{H}_2\text{O}$, which is a balanced equation. The convention in chemistry is to choose integer coefficients for reactants and products, so we'll multiply the equation by two to obtain the final answer

$$\text{CH}_4 + 2\text{O}_2 \;\to\; \text{CO}_2 + 2\text{H}_2\text{O}.$$

Balancing chemical equation may not seem like the most exciting technique ever, but it's a very useful skill to have when calculating things with chemistry. It's good to know that substances A and B can transform into substances C and D, but it's better to know *how much* of each reactant is consumed and how much of each product is produced per "unit of reaction." Once we've identified one "unit of reaction," we can calculate other quantities in terms of it, like measuring the energy released per unit of reaction. The combustion of 1[mol] ($= 6.022 \times 10^{23}$ molecules) of methane produces 890[kJ] of heat:

$$\text{CH}_4 + 2\text{O}_2 \;\to\; \text{CO}_2 + 2\text{H}_2\text{O} \;+\; 890\,[\text{kJ/mol}].$$

Now *this* is cool. If you're heating your chalet with methane gas, and you know how much joules of heat you'll need, then the balancing chemical equation will help you calculate how many litres of methane you need to stock to survive this winter.

Exercises

The exercises below aren't difficult, so you should totally try to solve them. Going through them will give you some extra practice with the Gauss–Jordan elimination procedure. It's been *ages* since Chapter 4 so a refresher can't hurt.

E8.1 Balance the chemical equation $\text{Al} + \text{O}_2 \;\to\; \text{Al}_2\text{O}_3$.

E8.2 Balance the equation $\text{Fe(OH)}_3 + \text{HCl} \;\to\; \text{FeCl}_3 + \text{H}_2\text{O}$.

8.2 Input–output models in economics

Suppose you're the top economic official of a small country and you want to make a production plan for the coming year. For the sake of simplicity, let's assume your country produces only three commodities: electric power, wood, and aluminum. Your job is to choose the production rates of these commodities: x_e, x_w, and x_a. The country must produce enough to satisfy both the internal demand and the external demand for these commodities. The problem is complicated because the production rates in one industry may affect the production rates of other industries. For instance, it takes some electric power to produce each unit of aluminum, so your production plan must account for both external demand for electric power, as well as *internal demand* for electric power for aluminum production. If there exist complex interdependences between the different internal industries, as is often the case, it can be difficult to pick the right production rates.

In reality, most high-ranking government officials base their decisions about which industry to sponsor based on the dollar amounts of the kickbacks and bribes they received during the previous year. Let's ignore reality for a moment and assume you're an honest economist interested in using math to do what is right for the country instead of abusing his/her position of power like a blood thirsty leech.

Let's assume the electric production x_e must satisfy an external demand of 25 units, plus an additional 0.05 units for each unit of wood produced (electricity needed for saw mill operations) and an additional 0.3 units for each unit of aluminum produced. The wood production must be 10 units plus additional small amounts that depend on x_e and x_a (wood for construction). The production of aluminum must match 14 units of external demand plus an additional 0.1 units for each unit of electric power (for repairs of electric cables). We can model the interdependence between the industries using the following system of equations:

$$\begin{aligned} x_e &= 25 + 0.05x_w + 0.3x_a \\ x_w &= 10 + 0.01x_e + 0.01x_a \\ x_a &= \underbrace{14}_{\text{external demand}} + \underbrace{0.1x_e}_{\text{internal demand}}. \end{aligned}$$

You can use linear algebra to solve this complicated industry interdependence problem, and choose appropriate production rates. Express

the system of equations as a matrix equation:

$$\underbrace{\begin{bmatrix} x_e \\ x_w \\ x_a \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} 25 \\ 10 \\ 14 \end{bmatrix}}_{\vec{d}} + \underbrace{\begin{bmatrix} 0 & 0.05 & 0.3 \\ 0.01 & 0 & 0.01 \\ 0.1 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_e \\ x_w \\ x_a \end{bmatrix}}_{\vec{x}}$$

This is known as a *Leontief input-output model* in honour of Wassily Leontief who first thought about applying linear algebra techniques to economics, and was awarded the Nobel prize for this contribution.

To find the appropriate production rates, we must solve for the unknown $\vec{x} = (x_e, x_w, x_a)$ in the above equation. The equation $\vec{x} = \vec{d} + A\vec{x}$ is a little unusual, but we can solve it using standard techniques:

$$\mathbb{1}\vec{x} = \vec{d} + A\vec{x} \quad \Rightarrow \quad (\mathbb{1} - A)\vec{x} = \vec{d} \quad \Rightarrow \quad \vec{x} = (\mathbb{1} - A)^{-1}\vec{d}.$$

For the case of the electricity, wood, and aluminum production scenario, the solution is $\vec{x} = (x_e, x_w, x_a) = (30.64, 10.48, 17.06)$. See **P4.9** for the details of the solution.

Note the electricity production rate is significantly higher than the external demand in order to account for the internal demand of electricity for the aluminum production. I'm not a big fan of economics and economists but I must admit this is a pretty neat procedure!

Links

[The Wikipedia article provides some historical context]
`http://en.wikipedia.org/wiki/Input-output_model`

8.3 Electric circuits

We can use Ohm's law to solve many circuit problems. Ohm's law, $V = IR$, tells us the voltage V required to "push" a current I through a resistor with resistance R. This simple equation is enough to solve for the currents and voltages in all circuit problems that involve batteries and resistors.

Since we're in "math land," the units of the circuit quantities won't play a direct role in our analysis, but it is still a good idea to introduce the units because units can help us perform dimensional analysis of the equations. Voltages are measured in volts [V], currents are measured in Amperes [A], and resistance is measured in Ohms [Ω]. Intuitively, the resistance of a resistor measures how difficult it is to "push" current through it. Indeed, the units for resistance have the dimensions of Volt per Ampere: $[\Omega] \equiv [\text{V/A}]$. The equation $V = RI$ tells us how much current I[A] flows through a resistor with resistance $R[\Omega]$ connected

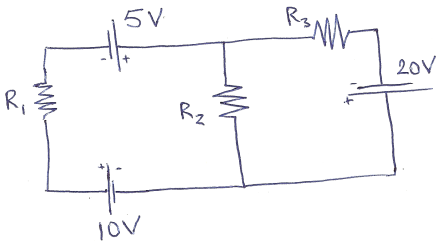
to a voltage source with potential V[V]. Alternatively, if we know the current I[A] and the resistance R[Ω], we can find V, the voltage applied to the resistor. A third way to use the equation $V = RI$ is to solve for the resistance R in cases when we know both V and I.

Example Your friend gives you a 121[Ω] light bulb (a resistor) and asks you to connect it outdoors on the backyard porch to provide some extra lighting for a summer party. You run to the basement and find three different spools of electric cable: a green one rated for currents of up to 0.5[A], a blue one rated for currents of up to 1[A], and a red spool of wire rated for currents of up to 2[A]. Knowing that the voltage coming out of the wall socket[1] is 110[V], what is the smallest-rating wire you can use to connect the light bulb?

A simple calculation using $V = RI$ shows the current that will flow in the wire is $I = \frac{V}{R} = \frac{110[\text{V}]}{121[\Omega]} = 0.909[\text{A}]$. Thanks to your calculation, you choose the blue wire rated for 1[A] knowing you won't have problems with wires overheating and causing a fire. Done. Now the only problem remaining is mosquito bites!

Given a complicated electric circuit in which several voltage sources (batteries) and resistors (light bulbs) are connected, it can be difficult to "solve for" all the voltages and currents in the circuit. Using the equation $V = RI$ for reach resistor leads to several equations that must be solved simultaneously to find the unknowns. Did someone say system of linear equations? Linear algebra to the rescue!

Knowing linear algebra will enable you to solve even the most complicated circuit using row operations (Gauss–Jordan elimination) in one or two minutes. We'll illustrate this application of linear algebra by solving the example circuit shown on the right, which involves three batteries (the parallel lines labelled + and −) and three resistors (the wiggly lines).



[1]The voltage coming out of wall outlets is actually not a constant 110[V] but a sine-wave oscillating between +155[V] and −155[V], but in this example we'll treat the socket's output as a constant voltage of 110[V].

Theory

Before we get started, let me introduce the minimum information you need to know about circuits: *Kirchhoff's voltage law* and *Kirchhoff's current law*.

The voltages in a circuit are related to the electric potential energy of the electrons flowing in the circuit. The electric potential is analogous to the concept of gravitational potential: a battery raises the electric potential of electrons like an elevator raises the gravitational potential of objects by increasing their height. Starting from this heightened potential, electrons will flow in the circuit and lose potential when passing through resistors. *Kirchhoff's voltage law* (KVL) states that the sum of the voltage gains and losses along any *loop* in the circuit must sum to zero. Intuitively, you can think of KVL as a manifestation of the *conservation of energy* principle: the potential gained by electrons when they pass through batteries must be lost in the resistors (in the form of heat). By the time the electrons complete their journey around any loop in the circuit, they must come back to their initial potential.

Kirchhoff's current law states that the total current flowing into a wire junction must equal the total current flowing out of the junction. You can think of this a manifestation of the *conservation of charge* principle: the total charge coming into a junction equals the total charge flowing out of the junction, because charges cannot be created or destroyed.

Using branch currents to solve the circuit

To "solve" a circuit is to find the currents that flow in each wire and the voltage across each resistor in the circuit. The first step is to define variables for each of the quantities of interest in the circuit as shown in Figure 8.1. We'll call I_1 the current that flows down through the middle wire of the circuit, which then splits into the current I_2 in the left branch and the current I_3 going to the right. Next we follow the currents in the circuit and label each resistor with "+" and "−" sides to indicate the direction of the voltage drop across it. The rule to follow is simple: the label "+" goes on the side where the current enters the resistor, and the label "−" goes on the side where the current leaves the resistor. This is because electric potential always *drops* when passing through a resistor.

We're now in a position to apply Kirchhoff's voltage and current laws to this circuit and obtain a set of equations that relate the unknown currents. Let's first apply Kirchhoff's voltage law to the loop along the path A-B-C-D-A, computing the total of the voltage gains

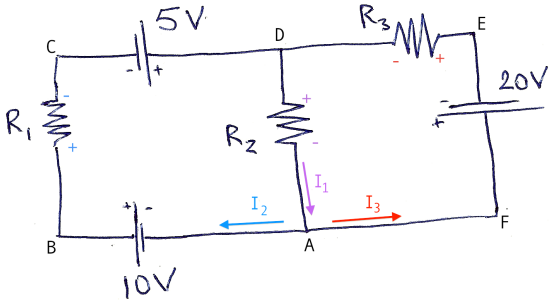


Figure 8.1: The circuit with branch currents labelled. Each resistor is assigned a *polarity* relative to the current flowing through it.

along this path:

$$+10 - R_1 I_2 + 5 - R_2 I_1 \;=\; 0.$$

Each battery produces a gain in potential for the electrons flowing through it. Each resistor leads to a drop in potential (negative gain) proportional to the current flowing through the resistor (recall $V = RI$).

Similarly, we obtain a second KVL equation by following the path A-F-E-D-A in the circuit:

$$-20 - R_3 I_3 - R_2 I_I \;=\; 0.$$

Note we're measuring voltage *gains* along this loop so we count the 20[V] battery as a negative gain (a voltage drop) since it is connected against the flow of current I_3.

We obtain a third equation linking the unknown currents from Kirchhoff's current law applied to junction A:

$$I_1 = I_2 + I_3.$$

Combining the three circuit equations, we obtain a system of three linear equations in three unknowns:

$$\begin{aligned} +10 - R_1 I_2 + 5 - R_2 I_1 &= 0, \\ -20 - R_3 I_3 - R_2 I_1 &= 0, \\ I_1 &= I_2 + I_3. \end{aligned}$$

Do you see where this is going? Perhaps rewriting the equations into the standard form we discussed in Section 4.1 will help you see what is going on:

$$\begin{aligned} -R_2 I_1 - R_1 I_2 \qquad\quad &= -15, \\ -R_2 I_1 \qquad\quad - R_3 I_3 &= 20, \\ I_1 \quad - I_2 \quad - I_3 &= 0. \end{aligned}$$

Rewriting the system of equations as an augmented matrix we obtain:

$$\left[\begin{array}{ccc|c} -R_2 & -R_1 & 0 & -15 \\ -R_2 & 0 & -R_3 & 20 \\ 1 & -1 & -1 & 0 \end{array}\right].$$

Assume the values of the resistors are $R_1 = 1[\Omega]$, $R_2 = 2[\Omega]$, and $R_3 = 1[\Omega]$. We substitute these values into the augmented matrix then find its reduced row echelon form:

$$\left[\begin{array}{ccc|c} -2 & -1 & 0 & -15 \\ -2 & 0 & -1 & 20 \\ 1 & -1 & -1 & 0 \end{array}\right] \xrightarrow{\text{RREF}} \left[\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 17 \\ 0 & 0 & 1 & -18 \end{array}\right].$$

The currents are $I_1 = -1[\text{A}]$, $I_2 = 17[\text{A}]$, and $I_3 = -18[\text{A}]$. The voltage drop across R_1 is 17[V] with polarity as indicated in the circuit. Negative currents indicate the current in the circuit flows in the opposite direction of the current label in the diagram. The voltage drop across R_2 and R_3 are 1[V] and 18[V] respectively, but with reverse polarity of the one indicated in the circuit. Use Figure 8.1 to verify these currents and voltages are consistent with the KVL equations we started from.

Using loop currents to solve the circuit

We'll now discuss an alternative approach for solving the circuit. In the previous section, we defined "branch current" variables and obtained two KVL equations and one KCL equation. Let's now solve the same circuit problem by defining "loop current" variables ad illustrated in Figure 8.2. We now define I_1 to be the current circulating in the left loop of the circuit, in the clockwise direction. Similarly, define I_2 to be the current in the right loop of the circuit, also flowing in the clockwise direction.

The analysis of the circuit using the loop currents is similar to what we saw above. The only tricky part is the voltage drop across the resistor R_2 through which both I_1 and I_2 flow. The voltage drop across the resistor is a linear function of the currents flowing through the resistor ($V = RI$), so we can analyze the effects of the two currents separately and then add the results. This is an instance of the *superposition principle* which is often used in physics. In the KVL equation for the left loop, the voltage drop across R_2 as the superposition of the voltage drop $-R_2I_1$ and the voltage gain R_2I_2, which combine to give the term $-R_2(I_1 - I_2)$. The opposite effects occur in the KVL equation for the right loop: I_2 causes a voltage drop $-R_2I_2$ while I_1 causes a voltage gain R_2I_1.

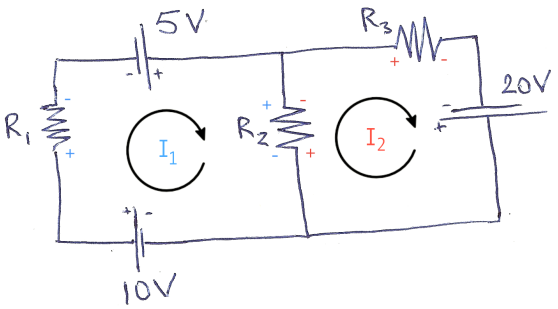


Figure 8.2: The circuit with "loop currents" labelled. Note the resistor R_2 has two currents flowing through it: I_1 downward and I_2 upward.

We thus obtain the following KVL equations:

$$\begin{aligned} +10 - R_1 I_1 + 5 - R_2(I_1 - I_2) &= 0 \\ +R_2(I_1 - I_2) - R_3 I_2 + 20 &= 0. \end{aligned}$$

We can rearrange this system of equations into the form:

$$\begin{aligned} (R_1 + R_2)I_1 - R_2 I_2 &= 15 \\ R_2 I_1 - (R_2 + R_3)I_2 &= -20, \end{aligned}$$

and then use standard linear algebra techniques to solve for the unknowns in a few seconds.

Using the same values for the resistors as given above ($R_1 = 1[\Omega]$, $R_2 = 2[\Omega]$, and $R_3 = 1[\Omega]$), we obtain the following solution:

$$\left[\begin{array}{cc|c} 3 & -2 & 15 \\ 2 & -3 & -20 \end{array}\right] \xrightarrow{\text{RREF}} \left[\begin{array}{cc|c} 1 & 0 & 17 \\ 0 & 1 & 18 \end{array}\right].$$

The loop currents are $I_1 = 17[\text{A}]$ and $I_2 = 18[\text{A}]$. This result is consistent with the result we obtained using branch currents.

Linear independence The abstract notion of *linear independence* manifests in an interesting way in electric circuit problems: we must choose the KVL equations that describe the current flowing in linearly independent loops. For example, there are actually three loops in the circuit from which we can obtain KVL equations: in addition to the two small loops we studied above, we can also apply KVL on the perimeter of the circuit as a whole: $+10 - R_1 I_1 + 5 - R_3 I_2 + 20 = 0$. It would seem then, that we have a system of *three* equations in two unknowns. However, the three equations are not independent: the KVM equation for the outer loop is equal to the sum of the KVL equations of the two small loops.

The procedures based on the "branch currents" and "loop currents" outlined above can be used to solve any electric circuit. For complicated circuits there will be a lot of equations, but using matrix methods you should be able to handle even the most complicated circuit. You can always use `SymPy` to do the RREF computation if it ever becomes too hairy to handle by hand.

Other network flows

The approach described above for finding the flow of currents in a circuit can be applied to many other problems with flows: the flow of cars on city streets, the flow of goods and services between economies, and the flow of information (data packets) through the Internet. In each case, a different law describes the flow in the network, but the same matrix techniques can be used to solve the resulting systems of linear equations.

8.4 Graphs

A *graph* is an abstract mathematical model that describes connections between a set of nodes. We call the nodes *vertices* and the connections *edges*. The graph is defined as pair of sets $G = (V, E)$, where V is the set of vertices and E is the set of edges in the graph. We can also describe the edges by specifying the *adjacency matrix* of the graph.

Rather than define graphs formally and in detail, we'll look at a simple graph example to give you an idea of main concepts and introduce graph notation. Figure 8.3 shows a small graph with five vertices and seven edges. This abstract link structure could represent many real-world scenarios: five websites and the hyperlinks between them, five Twitter accounts and their "following" relationships, or seven financial transactions between five businesses.

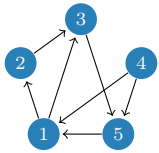


Figure 8.3: A simple graph with five vertices and seven edges.

The graph in Figure 8.3 is represented mathematically as $G = (V, E)$, where $V = \{1, 2, 3, 4, 5\}$ is the set of vertices, and $E = \{(1, 2), (1, 3), (2, 3), (3, 5), (4, 1), (4, 5), (5, 1)\}$ is the set of edges. Note the edge from vertex i to vertex j is represented as the pair (i, j).

Adjacency matrix

The *adjacency matrix* representation of this graph in Figure 8.3 is a 5×5 matrix A that contains the information about the edges in the graph. Specifically, $A_{ij} = 1$ if the edge (i, j) exists, otherwise $A_{ij} = 0$ if the edge doesn't exist:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Each row contains ones in the positions where edges exist. The adjacency matrix representation works in tandem with the integer labels of the vertices—Vertex 1 corresponds to the first row of A, Vertex 2 to the second row, and so on for the other rows. We don't need labels for the vertices since the labels can be deduced from their position in the matrix A.

Applications

The adjacency matrix of a graph can be used to answer certain questions about the graph's properties. For example, powers of the adjacency matrix A tell us information about the connectivity in the graph. The number ways to get from vertex i to vertex j in two steps is $(A^2)_{ij}$—the ij^{th} entry of A^2:

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

The number of ways to get from vertex i to vertex j in zero, one, or two steps is $\mathbb{1} + A + A^2$:

$$\mathbb{1} + A + A^2 = \begin{bmatrix} 1 & 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Observe that most of the graph G is well connected, except for Vertex 4, which has no inbound edges. The only way to get to Vertex 4 is if we start there.

The fact we can discuss graph connectivity by doing matrix algebra is amazing when you think about it! The entry $(\mathbb{1}+A+A^2)_{41} = 2$ tells us there are two ways to get from Vertex 4 to Vertex 1 in two steps or less. Indeed we can either transition directly through the edge $(4,1)$ in one step, or indirectly via Node 5 in two steps by passing through the edges $(4,5)$ and $(5,1)$. Rather than manually counting all possible paths between vertices, we can compute all possible paths at once using matrix algebra on the adjacency matrix A.

Discussion

The analysis of connectivity between vertices is used in many domains. Graphs are used to describe network flows, matching problems, social networks, webpages, and many other. In all these domains, the adjacency matrix plays a key role in graph representation. In Section 9.3 we'll study the graph of all webpages on the Web and discuss the Google's `PageRank` algorithm that uses the information in the adjacency matrix of the Web to compute an "importance" rank for each webpage.

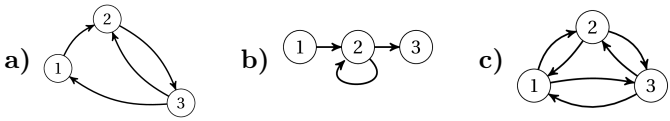
Links

[Graph theory and its applications]
`https://en.wikipedia.org/wiki/Graph_theory`

[More details about adjacency matrices with examples]
`https://en.wikipedia.org/wiki/Adjacency_matrix`

Exercises

E8.3 Find the adjacency matrix representation of the following graphs:



E8.4 For each of the graphs in **E8.3**, find the number of ways to get from vertex 1 to vertex 3 in two steps or less.

Hint: You can obtain the answer by inspection or by looking at the appropriate entry of the matrix $\mathbb{1} + A + A^2$.

8.5 Fibonacci sequence

We'll now look at a neat trick for computing the N^{th} term in the Fibonacci sequence $(0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots)$. The terms in the Fibonacci sequence $(a_0, a_1, a_2, \ldots)$ start with $a_0 = 0$, $a_1 = 1$, and then each subsequent term is computed as the sum of the two terms preceding it:

$$a_0 = 0, \quad a_1 = 1, \quad a_n = a_{n-1} + a_{n-2}, \text{ for all } n \geq 2.$$

You can apply this formula (the technical term for this type of formula is *recurrence relation*) to compute the N^{th} term in the sequence a_N. Using the formula to compute the 1000^{th} term in the sequence you'll have to do about 1000 steps of arithmetic to obtain a_{1000}. But do we really need N steps to compute a_N? In this section we'll learn an eigenvalue trick that allows us to compute a_N in just five step of symbolic computation, no matter how big N is!

First we express the recurrence relation as a matrix product:

$$\begin{aligned} a_{n+1} &= a_n + a_{n-1} \\ a_n &= a_n \end{aligned} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} a_{n+1} \\ a_n \end{bmatrix}}_{\vec{a}_n} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_n \\ a_{n-1} \end{bmatrix}}_{\vec{a}_{n-1}}.$$

We can compute the N^{th} term in the Fibonacci sequence by starting from the initial column vector $\vec{a}_0 = (a_1, a_0)^{\mathsf{T}}$, and repeatedly multiplying by the matrix A:

$$\begin{bmatrix} a_{n+1} \\ a_n \end{bmatrix} = A^N \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}.$$

We can "extract" a_N from the vector $\vec{a}_N$ by computing the dot product of $\vec{a}_N$ with the vector $(0, 1)$. This dot product operation has the effect of "selecting" the second entry of the vector $\vec{a}_N$.

Thus, we obtain the following compact formula for computing the N^{th} term in the Fibonacci sequence in terms of the N^{th} power of the matrix A:

$$a_N = (0, 1)A^N(1, 0)^{\mathsf{T}}.$$

Do you remember the eigendecomposition trick for computing powers of matrices by only computing powers of their eigenvalues? We can use the eigendecomposition trick to compute A^N very efficiently. The first step is to find the eigendecomposition of the matrix A:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ \frac{1}{\varphi} & -\varphi \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \varphi & 0 \\ 0 & \frac{-1}{\varphi} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \frac{5+\sqrt{5}}{10} & \frac{\sqrt{5}}{5} \\ \frac{5-\sqrt{5}}{10} & -\frac{\sqrt{5}}{5} \end{bmatrix}}_{Q^{-1}}.$$

Here $\lambda_1 = \varphi = \frac{1+\sqrt{5}}{2} \approx 1.618\ldots$ (the golden ratio) and $\lambda_2 = \frac{-1}{\varphi} = \frac{1-\sqrt{5}}{2} \approx -0.618\ldots$ (the negative inverse of the golden ratio) are the two eigenvalues of A. The columns of Q contain the corresponding eigenvectors of A.

We can compute A^N using the following formula:

$$A^N = \underbrace{AA\cdots A}_{N \text{ times}} = \underbrace{Q\Lambda Q^{-1}\ Q\Lambda Q^{-1}\cdots Q\Lambda Q^{-1}}_{N \text{ times}} = Q\Lambda^N Q^{-1}.$$

To compute A^N it is sufficient to compute the N^{th} powers of the eigenvalues $\lambda_1 = \varphi$ and $\lambda_2 = \frac{-1}{\varphi}$. For example, to compute a_5, the fifth element in the Fibonacci sequence, we compute A^5 using $Q\Lambda^5 Q^{-1}$, then use the formula $a_5 = (0,1)A^5(1,0)^\mathsf{T}$:

$$\begin{aligned} a_5 &= \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}^5\begin{bmatrix}1\\0\end{bmatrix} \\ &= \begin{bmatrix}0 & 1\end{bmatrix} Q \begin{bmatrix}\varphi^5 & 0\\ 0 & \frac{-1}{\varphi^5}\end{bmatrix} Q^{-1}\begin{bmatrix}1\\0\end{bmatrix} \\ &= \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}8 & 5\\ 5 & 3\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}8\\5\end{bmatrix} = 5. \end{aligned}$$

We can just as easily compute A^{55}:

$$A^{55} = Q\Lambda^{55}Q^{-1} = \begin{bmatrix}225851433717 & 139583862445\\ 139583862445 & 86267571272\end{bmatrix},$$

then compute a_{55} using the formula

$$a_{55} = (0,1)A^{55}(1,0)^\mathsf{T} = 139583862445.$$

Using the eigendecomposition trick allows us to take a "mathematical shortcut" and obtain the answer a_N in a constant number of math operations—regardless of the size of N. The steps are: compute the N^{th} power of Λ, then multiply Λ^N by Q^{-1} and $(1,0)^\mathsf{T}$ on the right, and by Q and $(0,1)$ on the left. This is interesting since other algorithms for computing the Fibonacci numbers usually take a number of steps proportional to the size of N.

There are some caveats to the approach outlined above. We assumed computing φ^N is a constant-time operation, which is not a realistic assumption on any computer. Also, infinite-precision (symbolic) manipulations are not realistic either because computers work with finite-precision approximations to real numbers, so the eigenvalue trick will not work for large N. The eigenvalue trick is only a theoretical result and can't be used in practical programs.

Links

[See the Wikipedia page for more on the Fibonacci numbers]
`https://en.wikipedia.org/wiki/Fibonacci_number`

Exercises

E8.5 Describe what happens to the ratio $\frac{a_{n+1}}{a_n}$ as $n \to \infty$.

Hint: Consider what happens to the powers of two eigenvalues λ_1^n and λ_2^n for large n.

E8.6 Compute the matrix products in the expression $(0,1)Q\Lambda^N Q^{-1}(1,0)^\mathsf{T}$ to obtain a closed form expression for a_N.

Hint:

8.6 Linear programming

In the early days of computing, computers were primarily used to solve optimization problems so the term "programming" is often used to describe optimization problems. *Linear programming* is the study of linear optimization problems that involve linear constraints. This type of optimization problems play an important role in business: the whole point of corporations is to constantly optimize profits, subject to time, energy, and legal constraints.

Many optimization problems can be expressed as *linear programs*:

$$\max_{x_1,x_2,\ldots,x_n} g(x_1,x_2,\ldots,x_n) \;=\; c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

subject to constraints:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2, \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m, \\ x_1 \geq 0, \quad x_2 \geq 0, \quad \ldots, \quad x_n &\geq 0. \end{aligned}$$

For example, the variables $x_1, x_2, \ldots, x_n$ could represent the production rates of n different products made by a company. The coefficients $c_1, c_2, \ldots, c_n$ represent the price of selling each product and $g(x_1, x_2, \ldots, x_n)$ represents the overall revenue. The m inequalities could represent various limitation of human resources, production capacity, or logistics constraints. We want to choose the production rates $x_1, x_2, \ldots, x_n$ that maximize the revenue, subject to the constraints.

The *simplex algorithm* is a systematic procedure for finding solutions to linear programming problems. The simplex algorithm is somewhat similar to the Gauss–Jordan elimination procedure since it uses row operations on a matrix-like structure called a *tableau*. For this reason, linear programming and the simplex algorithm are often inflicted upon students taking a linear algebra course, especially business students. I'm not going to lie to you and tell you the simplex algorithm is very exciting, but it is very powerful so you should know it exists, and develop a general intuition about how it works. Like with all things corporate-related, it's worth learning about it so you'll know the techniques of the enemy.

Since the details of the simplex algorithm might not be of interest to all readers of the book, this topic was split out as a separate tutorial, which you can read online at the link below.

[Linear programming tutorial]
`https://minireference.github.io/linear_programming/tutorial.pdf`

8.7 Least squares approximate solutions

An equation of the form $A\vec{x} = \vec{b}$ could have exactly one solution (if A is invertible), infinitely many solutions (if A has a null space), or no solution at all (if $\vec{b}$ is not in the column space of A). In this section we'll discuss the case with no solution and describe an approach for computing an *approximate* solution $\vec{x}^*$ such that the vector $A\vec{x}^*$ is as close as possible to $\vec{b}$.

We could jump right away to the formula for the least squares approximate solution ($\vec{x}^* = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b}$), but this would hardly be enlightening or useful for your understanding. Instead, let's learn about the least squares approximate solution in the context of a machine learning problem in which we'll try to predict some unknown quantities based on a linear model "learned" from past observations. This is called *linear regression* and is one of the most useful applications of linear algebra.

The database of current clients of your company contains all the information about the frequency of purchases f, value of purchases V, promptness of payment P, and other useful information. You know what is *really* useful information though? Knowing the customer lifetime value (CLV)—the total revenue this customer will generate during their entire relationship with your company. You have data on the CLVs of existing customers and you want to leverage this data to predict the CLVs of new customers.

You're given the profile parameters for N existing customers in the form of a vector $\vec{a}_i \equiv (f_i, V_i, P_i, \ldots)$ and calculated a customer life-

time value (CLV) for each existing customer $b_i \equiv CLV$. The *dataset* consists of observations $\vec{a}_i$ and outcomes b_i:

$$D = \left\{ \begin{bmatrix} — & \vec{a}_1 & — \\ — & \vec{a}_2 & — \\ — & \vec{a}_3 & — \\ & \vdots & \\ — & \vec{a}_N & — \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_N \end{bmatrix} \right\} = \{A, \vec{b}\}.$$

The clients' observational data is stored in an $N \times n$ matrix A and the corresponding CLVs are stored as a $N \times 1$ column vector $\vec{b}$.

Statement of the problem Given the $\vec{a}_k$ of a new customer, predict the customer's b_k, based on the information in the dataset D.

Linear model

A simple way to model the dependence of the label b_i on the observational data $\vec{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ is to use a linear model with n parameters $m_1, m_2, \ldots, m_n$:

$$y_{\vec{m}}(x_1, x_2, \ldots, x_n) = m_1 x_1 + m_2 x_2 + \cdots + m_n x_n = \vec{m} \cdot \vec{x}.$$

If the model is good then $y_{\vec{m}}(\vec{a}_i)$ approximates b_i well. But how should we measure the quality of the approximation?

Enter the error term that describes how the model's prediction $y_{\vec{m}}(\vec{a}_i)$ differs form the observed value b_i. The error term for the i^{th} customer is

$$e_i(\vec{m}) = |y_{\vec{m}}(\vec{a}_i) - b_i|^2.$$

The expression $e_i(\vec{m})$ measures the *squared* distance between the model's prediction and the known value. Our goal is to choose a model that makes the sum of all the error terms as small as possible:

$$S(\vec{m}) = \sum_{i=1}^{N} e_i(\vec{m}) = \sum_{i=1}^{N} |y_{\vec{m}}(\vec{a}_i) - b_i|^2 .$$

Intuitively, $S(\vec{m})$ is a good objective function to minimize because $S(\vec{m}) = 0$ if the model *perfectly* predicts the data. Any model prediction that overshoots or undershoots the correct b_i will be "penalized" by a large error term. Observe the objective function $S(\vec{m})$ is a quadratic function, so the "penalties" grow quadratically with the size of the discrepancy between the model and actual values.

Linear algebra formulation

When faced with an unfamiliar problem like finding a quadratically-penalized approximate solution to a system of equations $A\vec{x} = \vec{b}$ with no exact solution, you shouldn't be alarmed but stand your ground. Try to relate the problem to something you're more familiar with. Thinking about problems in terms of linear algebra can often unlock your geometrical intuition and show you a path toward the solution.

Using a matrix-vector product we can express the "vector prediction" of the model $\vec{y}_{\vec{m}}$ for the whole dataset in one shot: $\vec{y}_{\vec{m}} = A\vec{m}$. The "total squared error" function for the model $\vec{m}$ on the dataset $D = \{A, \vec{b}\}$ can be written as the following expression:

$$\begin{aligned} S(\vec{m}) &= \sum_{i=1}^{N} |y_{\vec{m}}(\vec{a}_i) - b_i|^2 \\ &= \|\vec{y}_{\vec{m}} - \vec{b}\|^2 \\ &= \|A\vec{m} - \vec{b}\|^2. \end{aligned}$$

The total squared error for the model $\vec{m}$ on the dataset $\{A, \vec{b}\}$ is the squared-length of the vector $A\vec{m} - \vec{b}$.

In the ideal case when the model perfectly matches the observations, the total squared error is zero and the equation $A\vec{m} = \vec{b}$ will have a solution:

$$A\vec{m} - \vec{b} = 0 \qquad \Rightarrow \qquad A\vec{m} = \vec{b}.$$

In practice, the model predictions $A\vec{m}$ will never perfectly match the data $\vec{b}$ so we must be content with approximate solution $\vec{m}$:

$$A\vec{m} \approx \vec{b}.$$

The *leasts-squares approximate solution* to this equation is chosen so as to minimize the total squared error function:

$$\min_{\vec{m}} S(\vec{m}) \quad = \quad \min_{\vec{m}} \left\| A\vec{m} - \vec{b} \right\|^2.$$

In other words, of all the possible approximate solutions $\vec{m}$, we must pick the one that makes the length of the vector $A\vec{m} - \vec{b}$ the smallest.

Finding the least-squares approximate solution

There are two possible approaches for finding the least-squares solution, denoted $\vec{m}^*$. We can either use calculus techniques to minimize the total squared error $S(\vec{m})$, or geometry techniques to find the shortest vector $(A\vec{m} - \vec{b})$.

Regardless of the approach chosen, the trick to finding the least-squares approximate solution to the equation $A\vec{m} \approx \vec{b}$ is to multiply the equation by A^T to obtain the equation:

$$A^\mathsf{T}A\,\vec{m} = A^\mathsf{T}\,\vec{b}.$$

The matrix $A^\mathsf{T}A$ will be invertible if the columns of A are linearly independent, which is the case for most tall-and-skinny matrices. We can therefore solve for $\vec{m}$ by multiplying by the matrix $(A^\mathsf{T}A)^{-1}$:

$$\vec{m} = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b}.$$

Indeed, this expression is the *least-squares solution* to the optimization problems we set out to solve in the beginning of this section:

$$\vec{m}^* = \underset{\vec{m}}{\operatorname{argmin}}\, S(\vec{m}) = \underset{\vec{m}}{\operatorname{argmin}} \left\| A\vec{m} - \vec{b} \right\|^2 = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b}.$$

Pseudoinverse The particular combination of A, its transpose A^T, and the inverse operation that we used to find the approximate solution $\vec{m}^*$ is called the *Moore–Penrose pseudoinverse* of the matrix A. We use the shorthand notation A^+ (not to be confused with $A^\dagger$), for the entire expression:

$$A^+ \equiv (A^\mathsf{T}A)^{-1}A^\mathsf{T}.$$

Applying A^+ to both sides of the approximate equation $A\vec{m} \approx \vec{b}$, is analogous to "solving" the equation by applying the inverse:

$$\begin{aligned} A^+A\vec{m} &= A^+\vec{b}, \\ \vec{m} &= A^+\vec{b}, \end{aligned}$$

but the solution is approximate since $A\vec{m} \neq \vec{b}$. The solution $A^+\vec{b}$ is optimal according to the total squared error criterion $S(\vec{m})$.

Example 1 Given the dataset with six samples (x_i, b_i): $(105, 45)$, $(113, 63)$, $(125, 86)$, $(137, 118)$, $(141, 112)$, $(153, 169)$, what is the best linear model $y_m(x) = mx$ for predicting b given x? The first step is to express the given samples in the standard form $\{A, \vec{b}\}$, where $A \in \mathbb{R}^{6\times1}$ and $\vec{b} \in \mathbb{R}^{6\times1}$. We then calculate the best parameter m using the Moore–Penrose inverse formula:

$$\{A, \vec{b}\} \equiv \left\{ \begin{bmatrix} 105 \\ 113 \\ 125 \\ 137 \\ 141 \\ 153 \end{bmatrix}, \begin{bmatrix} 45 \\ 63 \\ 86 \\ 118 \\ 112 \\ 169 \end{bmatrix} \right\} \quad \Rightarrow \quad m^* = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b} = 0.792.$$

The best fit linear model to the samples is the equation $y(x) = 0.792x$. For the details of the calculation, see `bit.ly/leastsq_ex1`. Figure 8.4 shows a scatter plot of the dataset $\{(x_i, b_i)\}$ and the graph of the best fit linear model through the data.

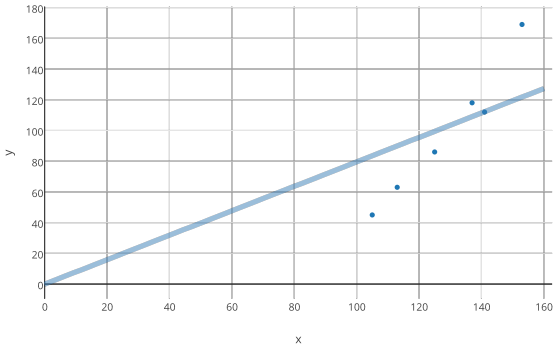


Figure 8.4: The "best fit" line of the form $y_m(x) = mx$ through a set of data points points $(x_i, b_i) \in \mathbb{R}^2$. The plot shows a scatter plot of the data points and the best fit line $y(x) = 0.792x$. The best fit line passes through the middle of the dataset and minimizes the sum of the squared vertical differences between the model output $y_m(x_i)$ and the true value b_i.

Geometric interpretation

The solution to the least squares optimization problem,

$$\vec{m}^* \quad = \quad \underset{\vec{m}}{\text{argmin}} \, \|A\vec{m} - \vec{b}\|^2,$$

can be understood geometrically as the search for the vector in the column space of A that is *closest* to the vector $\vec{b}$, as illustrated in Figure 8.5. As we vary the parameter vector $\vec{m}$, we obtain different vectors $A\vec{m} \in \mathcal{C}(A)$. Of all the points $A\vec{m}$ in the column space of A, the point $A\vec{m}^*$ is the closest to the point $\vec{b}$.

Let's define the "error vector" that corresponds to the difference between the model prediction $A\vec{m}$ and the actual value $\vec{b}$:

$$\vec{e} \quad \equiv \quad A\vec{m} - \vec{b}.$$

Using the geometric intuition from Figure 8.5, we see that the optimal solution $A\vec{m}^*$ occurs when the error vector is perpendicular to the column space of A. Recall the left-fundamental spaces of the matrix A: its column space $\mathcal{C}(A)$ and its orthogonal complement, the left null space $\mathcal{N}(A^\mathsf{T})$. Thus, if we want an error vector that is perpendicular to $\mathcal{C}(A)$, we must find an error vector that lies in the left null space

of A: $\vec{e}^* \in \mathcal{N}(A^\mathsf{T})$. Using the definition of left null space,

$$\mathcal{N}(A^\mathsf{T}) \equiv \{\vec{w} \in \mathbb{R}^N \mid \vec{w}^\mathsf{T} M = \vec{0}\},$$

we obtain the following equation that defines $\vec{m}^*$:

$$(\vec{e}^*)^\mathsf{T} A = 0 \qquad \Rightarrow \qquad (A\vec{m}^* - \vec{b})^\mathsf{T} A = 0.$$

Taking the transpose of the last equation we obtain $A^\mathsf{T}(A\vec{m}^* - \vec{b}) = 0$, which is equivalent to condition $A^\mathsf{T} A\vec{m}^* = A^\mathsf{T}\vec{b}$ used to find $\vec{m}^*$.

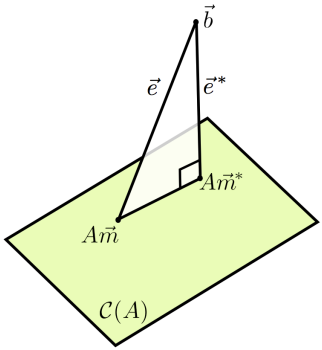


Figure 8.5: Linear regression can be seen as a "search" problem restricted to the column space of the data matrix A. The least-squares approximate solution $\vec{m}^* = A^+\vec{b}$ corresponds to the point in $\mathcal{C}(A)$ that is closest to $\vec{b}$. For this $\vec{m}^*$, the error vector $\vec{e}^*$ is perpendicular to the column space of A.

Using geometric intuition about vector spaces and orthogonality proves to useful for solving this complex optimization problem. Choosing $\vec{e}^*$ orthogonal to $\mathcal{C}(A)$, leads to the shortest vector $A\vec{m}^* - \vec{b}$, and produces the smallest total squared error $S(\vec{m}^*) = \|A\vec{m}^* - \vec{b}\|^2$.

Affine models

The Moore–Penrose pseudoinverse formula can be used to fit more complicated models. A simple extension of a linear model is the *affine model* $y_{\vec{m},c}(\vec{x}) = \vec{m} \cdot \vec{x} + c$, which adds a constant term c to the output of a linear model. The model parameters correspond to an $(n+1)$-vector $\vec{m}' = (c, m_1, m_2, \ldots, m_n)$:

$$y_{\vec{m}'}(x_1, x_2, \ldots, x_n) = m_0 + m_1 x_1 + m_2 x_2 + \cdots + m_n x_n = \vec{m}' \cdot (1, \vec{x}).$$

For uniformity of notation, we'll refer to the constant term as m_0 instead of c. You can think of the parameter m_0 as the y-intercept of the model.

To accommodate this change in the model, we must preprocess the dataset $D = \{A, \vec{b}\}$ to add a column of ones to the matrix A and turn it into an $N \times (n+1)$ matrix A'. The new dataset looks like this:

$$D' = \left\{ \begin{bmatrix} 1 & — & \vec{a}_1 & — \\ 1 & — & \vec{a}_2 & — \\ 1 & — & \vec{a}_3 & — \\ \vdots & & \vdots & \\ 1 & — & \vec{a}_N & — \end{bmatrix}, \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \right\} = \{A', \vec{b}\}.$$

Except for the preprocessing step which commonly called "data-massaging," the remainder of the steps for finding a least squares solution are the same as in the case of linear regression. We find the optimal parameter vector $\vec{m}'^*$ by applying the Moore–Penrose pseudoinverse formula:

$$\vec{m}'^* = (A'^{\mathsf{T}} A')^{-1} A'^{\mathsf{T}} \vec{b}.$$

Example 2 Find the best fit affine model $y_{\vec{m}'}(x) = m_0 + m_1 x$ to the data points from Example 1 (page 336). To find the best parameter vector $\vec{m}' = (m_0, m_1)^{\mathsf{T}}$, we first preprocess the dataset adding a column of ones to the data matrix then apply the pseudoinverse formula:

$$\{A', \vec{b}\} \equiv \left\{ \begin{bmatrix} 1 & 105 \\ 1 & 113 \\ 1 & 125 \\ 1 & 137 \\ 1 & 141 \\ 1 & 153 \end{bmatrix}, \begin{bmatrix} 45 \\ 63 \\ 86 \\ 118 \\ 112 \\ 169 \end{bmatrix} \right\} \quad \Rightarrow \quad \vec{m}'^* = A'^{+}\vec{b} = \begin{bmatrix} -210.4 \\ 2.397 \end{bmatrix}.$$

The best-fit affine model is $y(x) = -210.4 + 2.397x$. We omit the details of the matrix calculation for brevity, but you can verify everything is legit here: `bit.ly/leastsq_ex2`. See Figure 8.6 for the graph of the best-fit affine model.

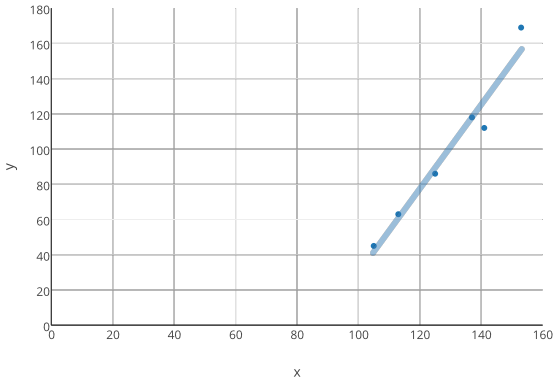


Figure 8.6: The best fit affine model $y_{\vec{m}'}(x) = m_0 + m_1 x$ through the data points is the line $y(x) = -210.4 + 2.397x$. Allowing for one extra parameter m_0 (the y-intercept), leads to a much better fitting model to the data, as compared to the fit in Figure 8.4.

Note the terms *linear regression* and *linear least squares* are used to refer to fitting both linear models $\vec{y}_{\vec{m}} = \vec{m} \cdot \vec{x}$, and affine models $\vec{y}_{\vec{m}'} = \vec{m}' \cdot (1, \vec{x})$. After all, an affine model is just a linear model with a y-intercept, so it makes sense to refer to them by the same name.

Quadratic models

Linear algebra techniques can also be used to find approximate solutions for nonlinear models. For the sake of simplicity, let's assume there are only two observed quantities in the dataset ($n = 2$). The most general quadratic model for two variables is:

$$y_{\vec{m}}(x, y) = m_0 + m_1 x + m_2 y + m_3 xy + m_4 x^2 + m_5 y^2.$$

The parameter vector is six-dimensional: $\vec{m} = (m_0, m_1, m_2, m_3, m_4, m_5)$.

Assuming we start from the following dataset

$$D = \left\{ \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \right\} = \{A, \vec{b}\}.$$

To use the Moore–Penrose pseudo-inverse formula, we must preprocess A by adding several new columns:

$$D' = \left\{ \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & x_N y_N & x_N^2 & y_N^2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \right\} = \{A', \vec{b}\}.$$

This preprocessing step allows us to compute a vector prediction of the model with parameters $\vec{m}$ on the entire dataset: $\vec{y}_{\vec{m}} \equiv A'\vec{m}$. The total squared error for this model is $S(\vec{m}) = \|\vec{y}_{\vec{m}} - \vec{b}\|^2$. The least squares approximate solution $\vec{m}^*$ is obtained as usual: $\vec{m}^* = (A'^{\mathsf{T}}A')^{-1}A'^{\mathsf{T}}\vec{b}$.

Note the number of parameters in a quadratic model grows—surprise, surprise—quadratically. There were $n = 2$ variables in the above example and the parameters vector $\vec{m}$ is six-dimensional. The the number of parameters of a general quadratic model in n variables is $\frac{1}{2}(n+1)(n+2) = \frac{1}{2}(n^2+3n+2)$. The number of parameters we need to "learn" is an important consideration to take into account when fitting models to large datasets.

Example 3 Imagine you're a data-friendly business person and you want to pick which model to *fit* to a dataset $\{A, b\}$ with $n = 1000$ features (the columns of A). You have the option of choosing an affine model $y_a(\vec{x}) = m_0 + m_1x_1 + \cdots + m_nx_n$, or a quadratic model $y_q(\vec{x}) = m_0 + m_1x_1 + \cdots + m_nx_n + m_{n+1}x_1y_1 + m_{n+2}x_1y_2 + \cdots$.

To fit an affine model, you'll need to preprocess $A \in \mathbb{R}^{N\times 1000}$ into $A' \in \mathbb{R}^{N\times 1001}$, then find the inverse of $(A'^{\mathsf{T}}A') \in \mathbb{R}^{1001\times 1001}$. That's totally doable. Probably in your web browser.

In contrast, the number of parameters in a quadratic model with $n = 1000$ dimensions is $\frac{1}{2}(1000+1)(1000+2) = 501501$. To fit a quadratic model you'll need to preprocess the data matrix to obtain $A' \in \mathbb{R}^{N\times 501501}$, then find the inverse of $(A'^{\mathsf{T}}A') \in \mathbb{R}^{501501\times 501501}$. That's a big matrix. You'll need one terabyte of memory just to store all the entries of the matrix $A'^{\mathsf{T}}A'$, and computing its inverse will take a *really* long time.

So the choice is made; affine model it is. Who cares if the model is less accurate than a quadratic model? If it works and provides value, you should use it. This is the reason why scientists, engineers, statisticians, and business folk are so crazy about building linear models, even though more advanced models are available. Linear models are the *sweet spot*: they have great modelling power, they're easy to implement, and they lead to computational problems that are easy to solve.

Links

[Further discussion about least squares problems on Wikipedia]
`https://en.wikipedia.org/wiki/Linear_regression`
`https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics)`

[More about the Moore–Penrose pseudoinverse]
`https://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse`

Exercises

E8.7 Calculate the total squared error $S(m^*) = \|Am^* - \vec{b}\|^2$ of the best fit linear model obtained in Example 1 (page 336).

Hint: The `Matrix` method `.norm()` might come in handy.

E8.8 Revisit Example 2 (page 339) and find the total squared error of the best-fit affine model $S(\vec{m}'^*) = \|A\vec{m}'^* - \vec{b}\|^2$.

8.8 Computer graphics

Linear algebra is the mathematical language of computer graphics. Whether you're building a simple two-dimensional game with stick figures or a fancy three-dimensional visualization, knowing linear algebra will help you understand the graphics operations that draw pixels on the screen.

In this section we'll discuss some basic computer graphics concepts. In particular, we'll introduce *homogenous coordinates* which are representations for vectors and matrices that use an extra dimension. Homogenous coordinates allow us to represent *all* interesting computer graphics transformations as matrix-vector products. We've already seen that scalings, rotations, reflections, and orthogonal projections can be represented as matrix-vector products; using homogenous coordinates we'll be able to represent translations and perspective projections as matrix products too. That's very convenient, since it allow the entire computer graphics processing "pipeline" to be understood in terms of a sequence of matrix multiplications.

Computer graphics is a vast subject and we don't have the space here to go into depth. To keep things simple, we'll focus on two-dimensional graphics, and only briefly touch upon three-dimensional graphics. The goal is not to teach you the commands of computer graphics APIs like OpenGL and WebGL, but to give you the basic math tools you'll need to understand what is going under the hood.

Affine transformations

In Chapter 6 we studied various linear transformation and their representation as matrices. We also briefly discussed the class of *affine transformations*, which consist of a linear transformation followed by a translation

$$\vec{w} = T(\vec{v}) + \vec{d}.$$

In the above equation, the input vector $\vec{v}$ is first acted upon by a linear transformation T then the output of T is translated by the displacement vector $\vec{d}$ to produce the output vector $\vec{w}$.

In this section we'll use *homogenous coordinates* for vectors and transformations, which allow us to express affine transformations as a matrix-vector product in a larger vector space:

$$\vec{w} = T(\vec{v}) + \vec{d} \qquad \Leftrightarrow \qquad \vec{W} = A\vec{V}.$$

If $\vec{v}$ is an n-dimensional vector, then its representation in homogenous coordinates $\vec{V}$ is an $(n+1)$-dimensional vector. The $(n+1) \times (n+1)$ matrix A contains the information about both the linear transformation T and the translation $\vec{d}$.

Homogenous coordinates

Instead of using a triple of cartesian coordinates to represent points $p = (x, y, z)_c \in \mathbb{R}^3$, we'll use the quadruple $P = (x, y, z, 1)_h \in \mathbb{R}^4$, which is a representation of the same point in *homogenous coordinates*. Similarly, the vector $\vec{v} = (v_x, v_y, v_z)_c \in \mathbb{R}^3$, corresponds to the four-vector $\vec{V} = (v_x, v_y, v_z, 1)_h \in \mathbb{R}^4$ in homogenous coordinates. Though there is no mathematical difference between points and vectors, we'll stick to the language of points as it is more natural for graphics problems. Homogenous coordinates of the form $(x, y, z, 0)_h$ correspond to points at infinity in the direction of $(x, y, z)_c$.

An interesting property of homogenous coordinates is that they're not unique. The vector $\vec{v} = (v_x, v_y, v_z)_c$ corresponds to a whole *set of point* in homogenous coordinates: $\vec{V} = \{(\alpha v_x, \alpha v_y, \alpha v_z, \alpha)_h\}$, for $\alpha \in \mathbb{R}$. This makes homogenous coordinates invariant to scaling:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}_c \qquad \Leftrightarrow \qquad \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}_h = \begin{bmatrix} 5a \\ 5b \\ 5c \\ 5 \end{bmatrix}_h = \begin{bmatrix} 500a \\ 500b \\ 500c \\ 500 \end{bmatrix}_h.$$

This is kind of weird, but this extra freedom to rescale vectors arbitrarily will lead to many useful applications.

To convert from homogenous coordinates $(X, Y, Z, W)_h = (a, b, c, d)_h$ to cartesian coordinates, we divide each component by the W-component to obtain the equivalent vector $(X, Y, Z, W)_h = (\frac{a}{d}, \frac{b}{d}, \frac{c}{d}, 1)_h$, which corresponds to the point $(x, y, z)_c = \left(\frac{a}{d}, \frac{b}{d}, \frac{c}{d}\right)_c \in \mathbb{R}^3$.

In the case when the underlying cartesian space is two-dimensional, the point $p = (x, y)_c \in \mathbb{R}^2$ is written as $P = (X, Y, W)_h = (x, y, 1)_h$ in homogenous coordinates. The homogenous coordinates $(X, Y, W)_h = (a, b, d)_h$ with $d \neq 0$ represent the point $(x, y)_c = \left(\frac{a}{d}, \frac{b}{d}\right)_c \in \mathbb{R}^2$. You can visualize the conversion Figure 8.7 illustrates how the plane $w = 1$ is used to obtain the cartesian coordinates.

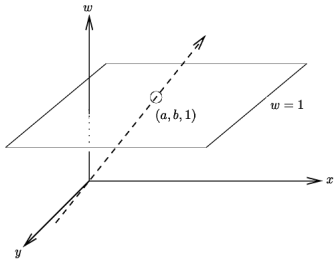


Figure 8.7: Two-dimensional points and vectors correspond to infinite lines in homogenous coordinates. The cartesian coordinates (a, b) can be identified with a "representative" point on the infinite line. By convention we represent this point as $(a, b, 1)$ in homogenous coordinates, which corresponds intersection of the infinite line with plane $w = 1$.

To distinguish cartesian vectors from homogenous vectors, we'll use a capital letters like $P, \vec{A}, \vec{B}, \ldots$ for points and vectors in homogenous coordinates, and lowercase letters like $p, \vec{a}, \vec{b}, \ldots$ when referring to vectors in cartesian coordinates.

Affine transformations in homogenous coordinates Consider the affine transformation that consists of the transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ followed by a translation by $\vec{d} = (d_1, d_2)$. If T is represented by the matrix $M_T = \left[\begin{smallmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{smallmatrix}\right]$, then the affine transformation as a whole can be represented as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & d_1 \\ m_{21} & m_{22} & d_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

As you can see, there's nothing fancy about homogenous coordinates; we wanted to be able to add constants terms to each component of the vector so we invented an extra "constant" component to each vector and an extra column that "hits" these constants. Make sure you can trace the above matrix-vector products and convince yourself there is no new math—just the good old matrix-vector product you're familiar with.

Homogenous coordinates and projective geometry are powerful mathematical techniques with deep connections to many advanced math subjects. For the purpose of this appendix we can only give an overview from an "engineering" perspective—what can you do with matrix-vector products in homogenous coordinates?

Graphics transformations in 2D

This "extra dimension" in the homogenous coordinates representation for vectors and points allows us to express most geometric transformation the form $T : \mathbb{R}^2 \to \mathbb{R}^2$ in a very succinct manner. In addition to the affine transformations described above, homogenous coordinates also allow us to perform *perspective transformations*, which are of central importance in computer graphics. The most general transformation we can perform using homogenous coordinates is

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & d_1 \\ m_{21} & m_{22} & d_2 \\ p_1 & p_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

where $M = \left[\begin{smallmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{smallmatrix}\right]$ corresponds to any linear transformation, $\vec{d} = (d_1, d_2)$ corresponds to a translation, and (p_1, p_2) is a perspective transformation.

Linear transformations

Let R_θ be the clockwise rotation by the angle θ of all points in $\mathbb{R}^2$. In homogenous coordinates, this rotation is represented as

$$p' = R_\theta(p) \quad \Leftrightarrow \quad P' = M_{R_\theta} P,$$

where M_{R_θ} is the following 3×3 matrix:

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Indeed the 2×2 top-left entries of matrices in homogenous coordinates can be used to represent any linear transformation: projections, reflections, scalings, and shear transformations. The following equation shows the homogenous matrices the reflection through the x-axis M_{R_x}, an arbitrary scaling M_S, and a shear along the x-axis M_{SH_x}:

$$M_{R_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_{SH_x} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Orthogonal projections

Projections can also be represented as 3×3 matrices. The projection onto the x-axis corresponds to the following representation in homogenous coordinates:

$$p' = \Pi_x(p) \qquad \Leftrightarrow \qquad \begin{bmatrix} x' \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Translation

The translation by the displacement vector $\vec{d} = (d_x, d_y)$ corresponds to the matrix

$$M_{T_{\vec{d}}} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}.$$

Note the identity transformation in the top-left part the matrix: we don't want any linear transformation performed—just a translation.

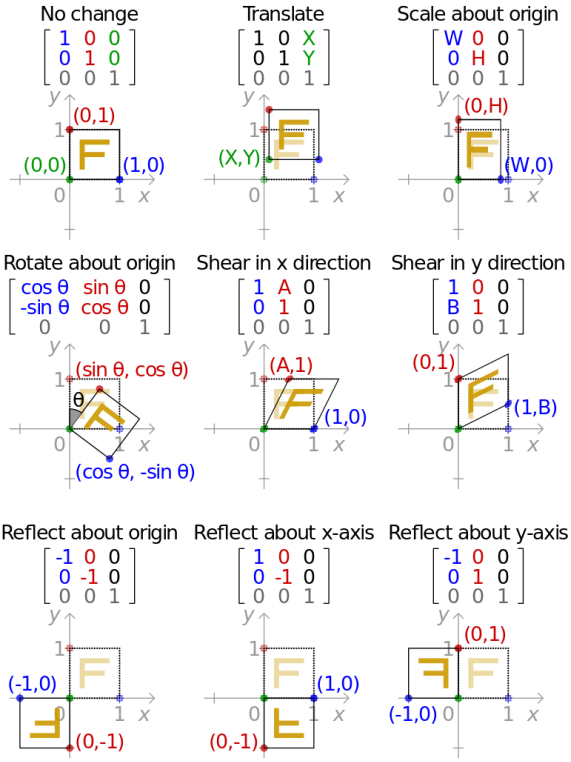


Figure 8.8: Illustration of the different transformations on a sample shape. source: `wikipedia File:2D_affine_transformation_matrix.svg`

Rotations, reflections, shear transformations, and translations can all be represented as multiplications by appropriate 3×3 matrices. Figure 8.8 shows examples of transformations that we can perform using the matrix-vector product in homogenous coordinates. I hope by now you're convinced that this idea of adding an extra "constant" dimension to vectors, is a useful thing to do.

But wait, there's more! We haven't even seen yet what happens when we put coefficients in the last row of the matrix.

Perspective projections

The notion of perspective comes from the world of painting. For a painting to look "realistic," objects' relative distance from the viewer must be conveyed by their different size. Distant objects in the scene are drawn smaller than objects in the foreground. Rather than give you the general formula for perspective transformations, we'll derive the matrix representation for perspective transformations from first principles. Trust me, it will be a lot more interesting.

We can understand perspective transformations by tracing out imaginary light rays that start from some object and go toward the eye of an observer O. The image of the perspective projections is where the light ray hits the *projective plane*. Since we're working in $\mathbb{R}^2$, we can draw a picture of what is going on.

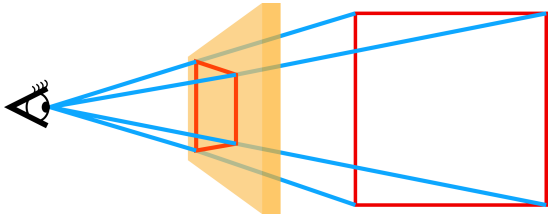


Figure 8.9: The perspective projection of a square onto a screen. Observe that the side of the square that is closer to the observer appear longer than the side that is farther.

Suppose we want to compute the *perspective transformation* to the line with equation $y = d$ for an observer placed at the origin $O = (0, 0)$. Under this perspective projection, every point $p = (x, y)$ in the plane maps to a point $p' = (x', y')$ on the line $y = d$. Figure 8.10 illustrates the situation.

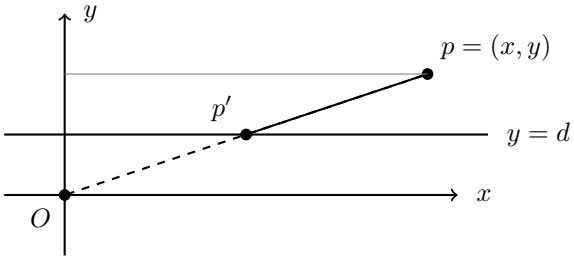


Figure 8.10: The point $p' = (x', y')$ is the projection of the point $p = (x, y)$ onto the plane with equation $y = d$. The ratio of lengths d/y must equal the ratio of lengths x'/x.

The only math prerequisite you need to remember is the general principle for *similar* triangles. If the triangle with sides a, b, c is similar to a triangle with sides a', b', c', this means the ratios of the sides' lengths are equal:

$$\frac{a'}{a} \quad = \quad \frac{b'}{b} \quad = \quad \frac{c'}{c}.$$

Since the two triangles in Figure 8.10 are similar, we know $\frac{x'}{x} = \frac{d}{y}$ and therefore directly obtain the expression for (x', y') as follows:

$$x' = \frac{d}{y}x,$$
$$y' = \frac{d}{y}y = d.$$

This doesn't look very promising so far since the expression for x' contains a division by y. The set of equations is not linear in y and therefore cannot be express as a matrix-product. If only there was some way to represent vectors and transformations that **also allows division** by coefficients too.

Let's analyze the *perspective transformation* in terms of homogenous coordinates, and see if something useful comes out:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{d}{y}x \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{y}{d} \end{bmatrix}.$$

The second equality holds because vectors in homogenous coordinates are invariant to scalings: $(a, b, c)_h = \alpha(a, b, c)_h$ for all α. We can shift the factor $\frac{d}{y}$ as we please: $(\frac{d}{y}x, d, 1) = \frac{d}{y}(x, y, \frac{y}{d}) = (x, y, \frac{y}{d})$. In the alternate homogenous coordinates expression, we're no longer dividing by y. This means we can represent the perspective transformation as a matrix-vector product:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{y}{d} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Now that's interesting. By preparing the vector (X', Y', W') with a third component $W' \neq 1$, we can force each coefficient to be scaled by $\frac{1}{W'}$, which is exactly what we need for perspective transformations. Depending on how the coefficient W' is constructed, different perspective transformations can be obtained.

A *perspective projection* transformation is a perspective transformation followed by an orthogonal projection that deletes some of the

vector's components. The perspective projection onto the line with equation $y = d$ is the composition of a *perspective transformation* $P : \mathbb{R}^3 \to \mathbb{R}^3$ followed by an orthogonal projection $\Pi_x : \mathbb{R}^3 \to \mathbb{R}^2$ which simply discards the y' coordinate:

$$\begin{bmatrix} x' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{d} & 0 \end{bmatrix}}_{\Pi_x P} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Note only the x' coordinate remains after the projection. This is the desired result since we want only the "local" coordinates of the projection plane $y = d$ to remain after the projection.

Certain textbooks on computer graphics discuss only the combined perspective-plus-projection transformation $\Pi_x P$, as described on the right side of the above equation. We prefer treat the perspective transformation separately from the projection since it makes the math easier to understand. Also, in many practical applications, keeping the "depth information" of the different objects which we're projecting (the y-coordinates) can be useful for determining which objects appear in front of others.

General perspective transformation

Let's now look at the general case of a perspective transformation that projects arbitrary points $p = (x, y)$ onto the line $ax + by = d$. Again, we assume the observer is located at the origin $O = (0, 0)$. We want to calculate the coordinates of the projected point $p' = (x', y')$, as illustrated in Figure 8.11.

The reasoning we'll use to obtain the general perspective transformation is similar to the special case we considered above, and also depends on the similar triangles. Define α to the projection of the point p onto the line with direction vector $\vec{n} = (a, b)$ passing through the origin. Using the general formula for distances (see Section 5.1), we can obtain the length ℓ from O to α:

$$\ell \;\equiv\; d(O, \alpha) = \frac{\vec{n} \cdot p}{\|\vec{n}\|} = \frac{ax + by}{\|\vec{n}\|}.$$

Similarly, we define ℓ' to be the length of the projection of p' onto $\vec{n}$:

$$\ell' \;\equiv\; d(O, \alpha') = \frac{\vec{n} \cdot p'}{\|\vec{n}\|} = \frac{ax' + by'}{\|\vec{n}\|} = \frac{d}{\|\vec{n}\|}.$$

The last equation holds because the point p' is on the line with equation $ax + by = d$.

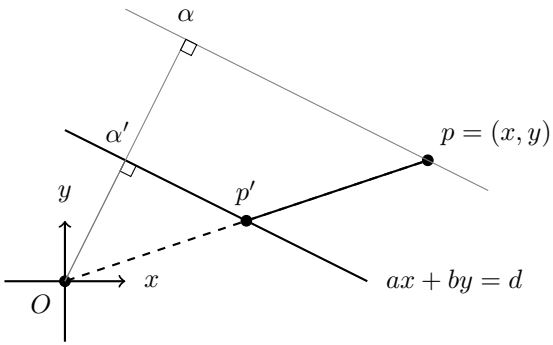


Figure 8.11: The point $p' = (x', y')$ is the projection of the point $p = (x, y)$ onto the line with equation $ax + by = d$. We define points α' and α in the direction of line's normal vector $\vec{n} = (a, b)$. The distances from the origin to these points are ℓ' and ℓ respectively. We have $\ell'/\ell = x'/x = y'/y$.

By the similarity of triangles, we know the ratio of lengths x'/x and y'/y must equal the ratio of orthogonal distances ℓ'/ℓ:

$$\frac{x'}{x} = \frac{y'}{y} = \frac{\ell'}{\ell} = \frac{d}{ax + by}.$$

We can use this fact to express the coordinates x' and y' in terms of the original x and y coordinates. As in the previous case, expressing points in homogenous coordinates allows us to arbitrarily shift scaling factors:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\ell'}{\ell}x \\ \frac{\ell'}{\ell}y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{\ell}{\ell'} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{ax+by}{d} \end{bmatrix}.$$

The last expression is linear in the variables x and y, therefore it has a matrix representation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{a}{d} & \frac{b}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

This is the most general perspective transformation. The "scaling factor" W' is a linear combination of the input coordinates: $W' = \frac{a}{d}x + \frac{b}{d}y$. Observe that setting $a = 0$ and $b = 1$ we recover the perspective transformation for the projection onto line $y = d$.

Graphics transformations in 3D

Everything we saw in the previous section about two-dimensional transformations also applies to three-dimensional transformations. A three-dimensional cartesian coordinate triple $(x, y, z)_c \in \mathbb{R}^3$ is represented as $(x, y, z, 1)_h \in \mathbb{R}^4$ in homogenous coordinates. Transformations in homogenous coordinates are represented by 4×4 matrices. For example the most general affine transformation corresponds to

$$A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & d_1 \\ m_{21} & m_{22} & m_{23} & d_2 \\ m_{31} & m_{32} & m_{33} & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the perspective transformation onto the line $ax + by + cz = d$ with the observer at the origin is

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{a}{d} & \frac{b}{d} & \frac{c}{d} & 0 \end{bmatrix}.$$

These should look somewhat familiar to you. Indeed, except for the larger space and additional degrees of freedom, the mechanics of using homogenous coordinates is the same in 3D as in 2D.

The best past of using the homogenous coordinates representation for all transformations is that we can easily compose them together, like LEGOs. We'll learn about this in the next section.

3D graphics programming

Inside every modern computer there is a special-purpose processor dedicated to computer graphics operations called the *graphics processing unit* (GPU). Modern GPUs can have thousands of individual graphics processing units called *shaders* and each shader can perform millions of linear algebra operations per second. Think about it, thousands of processors working in parallel doing matrix-vector products for you—that's a lot of linear algebra calculating power!

The reason why we need so much processing power is because 3D models are made up of thousands of little polygons. Drawing a 3D scene, also known as *rendering*, involves performing linear algebra manipulations on all these polygons. This is where the GPU comes in. The job of the GPU is to translate, rotate, and scale the polygons of the 3D models placing them into the scene, and then compute what the scene looks like when projected to a two-dimensional window through which you're observing the virtual world. This transformation from the model coordinates, to world coordinates, and then

to screen coordinates (pixels) is carried out in a *graphics processing pipeline*.

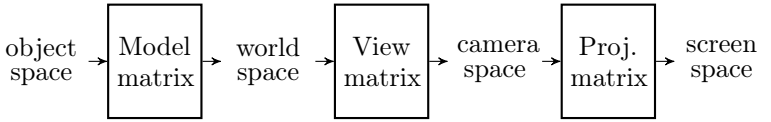


Figure 8.12: A graphics processing pipeline for drawing 3D objects on the screen. A 3D model is composed of polygons expressed with respect to a coordinate system centred on the object. The model matrix positions the object in the scene, the view matrix positions the camera in the scene, and finally the projection matrix computes what should appear on the screen.

We can understand the graphics processing pipeline as a sequence of matrix transformations: the model matrix M, the view matrix V, and the projection matrix Π_s. The GPU applies this sequence of operations to each of the object's vertices $(x, y, z, 1)_o$, to obtain the pixel coordinates of the vertices on the screen $(x', y')_s$:

$$\begin{bmatrix} x' \\ y' \end{bmatrix}_s = \Pi_s V M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_m \quad \Rightarrow \quad (x, y, z, 1)_m M^\mathsf{T} V^\mathsf{T} \Pi_s^\mathsf{T} = (x', y')_s.$$

It is customary to represent the graphics processing pipeline in the "transpose picture" so that data flows from left to right. All vectors are row vectors and we use the transpose of each operation in the pipeline.

It is not necessary to compute three matrix-vector products for each vertex. It is much more efficient to compute a combined transformation matrix $C^\mathsf{T} = M^\mathsf{T} V^\mathsf{T} \Pi_s^\mathsf{T}$, then apply the combined matrix C^T to each of the coordinates of the 3D object. Similarly, when drawing another object, only the model matrix needs to be modified, while the view and projection matrices remain the same.

Practical considerations

We discussed homogenous coordinates and the linear algebra transformations used for computer graphics. This is the essential "theory" you'll need to get started with computer graphics programming. The graphics pipeline used in modern 3D software has a few more steps than the simplified version we showed in Figure 8.12. We'll now discuss some practical considerations in point form.

- There are actually *two* graphics pipelines at work in the GPU. The *geometry pipeline* handles the transformation of polygons

that make up the 3D objects. A separate *texture pipeline* controls the graphics pattern that polygons will be filled with. The final step in the rendering process combines the outputs of the two pipelines.

- The Model and View matrices can be combined to form a ModelView matrix that converts from object to camera coordinates.
- Not all objects in the scene need to be rendered. We don't need to render objects that fall outside of the camera's viewing angle. Also we we can skip objects that are closer than the *near plane* or farther than the *far plane* of the camera. Though the scene could have infinite extent, we're only interested in rendering the subset of the scene that we want displayed on the screen, which we call the *view frustum*. See the illustration in Figure 8.13.

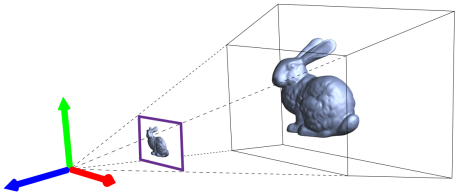


Figure 8.13: The perspective transformation used to project a three-dimensional model onto the two-dimensional screen surface. Note only a subset of the scene (the *view frustum*) is drawn, which is cut between the *near plane* and the *far plane* within the field of view.

I encourage your to pursue the subject of computer graphics further on your own. There are excellent free resources for learning OpenGL and WebGL on the web.

Discussion

Homogenous coordinates have many other applications that we did not have time to discuss. In this section we'll briefly mention some of these with the hope of inspiring you to research the subject further.

Homogenous coordinates are very convenient for representing planes in $\mathbb{R}^3$. The plane with general equation $ax+by+cz = d$ is represented by the four-vector $\vec{N} = (a, b, c, -d)$ in homogenous coordinates. This is a natural thing to do—instead of describing the plane's normal vector $\vec{n}$ separately from the constant d, we can represent the plane by an "enhanced" normal vector $\vec{N} \in \mathbb{R}^4$:

$$\text{Plane with } \vec{n} = (a, b, c) \text{ and constant } d \quad \Leftrightarrow \quad \vec{N} = (a, b, c, -d).$$

This is really cool because we can now use the same representation for both vectors and planes and perform vector operations between them. Yet again we find a continuation of the "everything is a vector" theme that we've seen throughout the book.

What good is representing planes in homogenous coordinates? How about being able to "check" whether any point $\vec{P}$ lies in the plane $\vec{N}$ by computing the dot product $\vec{N} \cdot \vec{P}$? The point $\vec{P}$ lies inside the plane with normal vector $\vec{N}$ if and only if $\vec{N} \cdot \vec{P} = 0$. Consider the point $\vec{P}_1 = (0, 0, \frac{d}{c}, 1)$ that lies in the plane $ax + by + cz = d$, we can verify that

$$\vec{N} \cdot \vec{P}_1 = (a, b, c, -d) \cdot (0, 0, \tfrac{d}{c}, 1) = 0.$$

It's also possible to easily obtain the homogenous coordinates of the plane $\vec{N}$ that passes through any three points $\vec{P}$, $\vec{Q}$, and $\vec{R}$. We're looking for a vector $\vec{N}$ that is perpendicular to all three points. We can obtain $\vec{N}$ using a generalization of the the cross product, computed using a four-dimensional determinant:

$$\vec{N} = \begin{vmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 & \hat{e}_4 \\ p_x & p_y & p_z & p_w \\ q_x & q_y & q_z & q_w \\ r_x & r_y & r_z & r_w \end{vmatrix},$$

where $\{\hat{e}_1, \hat{e}_2, \hat{e}_3, \hat{e}_4\}$ is the standard basis for $\mathbb{R}^4$. Yes, seriously. I bet you haven't seen a four-dimensional cross product before. This stuff is wild! See the links below if you want to learn more about homogenous coordinates, projective spaces, or computer graphics.

Links

[Homogeneous coordinates lecture notes]
`http://web.cs.iastate.edu/~cs577/handouts/homogeneous-coords.pdf`

[Tutorials about OpenGL programming in `C++`]
`http://www.songho.ca/opengl/index.html`

[Tutorials about WebGL from the Mozilla Developer Network]
`https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API`

[Another detailed tutorial series on WebGL]
`https://github.com/greggman/webgl-fundamentals/`

8.9 Cryptography

Cryptography is the study of secure communication. The two main tasks that cryptographers aim to achieve are private communication

(no eavesdroppers) and authenticated communication (no impersonators). Using algebraic operations over finite fields $\mathbb{F}_q$, it's possible to achieve both of these goals. Math is the weapon for privacy!

The need for private communication between people has been around since long before the development of modern mathematics. Thanks to modern mathematical techniques, we can now perform cryptographic operations with greater ease, and build crypto systems whose security is guaranteed by mathematical proofs. In this section we'll discuss the famous *one-time pad* encryption technique invented by Claude Shannon. One-time pad encryption is a *provably* secure crypto system. In order to understand what that means precisely, we'll first need some context about what is studied in cryptography.

Context

The secure communication scenarios we'll discuss in this section involve three parties:

- Alice is the message sender
- Bob is the message receiver
- Eve is the eavesdropper

Alice wants to send a private message to Bob, but Eve has the ability to see all communication between Alice and Bob. You can think of Eve as a `facebook` administrator, or a employee of the Orwellian, privacy-invading web application *du jour*. To defend against Eve, Alice will *encrypt* her messages before sending them to Bob, using a secret key only Alice and Bob have access to. Eve will be able to capture the encrypted messages (called *ciphertexts*) but they will be unintelligible to her because of the encryption. Assuming Bob receives the messages from Alice, he'll be able to *decrypt* them using his copy of the secret key. Using encryption allows Alice and Bob to have *private* communication despite Eve's eavesdropping.

Cryptography is an interesting subject that is very pertinent in the modern age of pervasive network surveillance. It's a bit like learning kung fu for self defence. You don't need to go around beating up people, but you should know how to defend yourself in case something comes up. I encourage you to learn more about practical cryptography for your communications. You don't need to stare at command-line terminals with green letters scrolling on them like in the matrix, but knowing some basics about passwords, secret keys, digital signatures, and certificates will really help you keep up with the forces out there.

The material in this section only scratches the surface of all there is to learn in cryptography and is only intended to illustrate the main

concepts through a simple example: the *one-time pad* encryption protocol. This simple encryption scheme is provably, unconditionally secure, given some assumptions about the secret key $\vec{k}$.

Definitions

A cryptographic protocol consists of an encryption function, a decryption function, and a procedure for generating the secret key $\vec{k}$. For simplicity we assume messages and keys are all binary strings:

- $\vec{m} \in \{0,1\}^n$: the *message* or *plaintext* is a bitstring of length n
- $\vec{k} \in \{0,1\}^n$: the *key* is a shared secret between Alice and Bob
- $\vec{c} \in \{0,1\}^n$: the *ciphertext* is the encrypted message
- $\text{Enc}(\vec{m}, \vec{k})$: the encryption function that takes as input a message $\vec{m} \in \{0,1\}^n$ and the key $\vec{k} \in \{0,1\}^n$ and produces a ciphertext $\vec{c} \in \{0,1\}^n$ as output
- $\text{Dec}(\vec{c}, \vec{k})$: the decryption function that takes as input a ciphertext $\vec{c} \in \{0,1\}^n$ and the key $\vec{k} \in \{0,1\}^n$ and produces the decrypted message $\vec{m} \in \{0,1\}^n$ as output

We consider the protocol to be secure if Eve cannot gain any information about the messages $\vec{m}_1, \vec{m}_2, \ldots$ from the ciphertexts $\vec{c}_1, \vec{c}_2, \ldots$ she intercepts.

Before we describe the one-time pad encryption protocol and discuss its security, we must introduce some background about binary numbers.

Binary

The cryptographic operations we'll discuss in this section apply to information encoded in *binary*. A *bit* is an element of the binary field, a finite field with two elements $\mathbb{F}_2 \equiv \{0,1\}$. A *bitstring* of size n is an n-dimensional vector of bits $\vec{v} \in \{0,1\}^n$. For notational simplicity, we denote bitstrings as $v_1 v_2 \cdots v_n$ instead of using the usual vector notation $(v_1, v_2, \ldots, v_n)$. For example, 0010 is a bitstring of length 4.

Information in modern digital systems is encoded in binary. Every file on your computer consists of long sequences of bits that are copied between disk memory, RAM, caches, and CPU registers. For example, using the ASCII encoding convention the character "a" is encoded as the bitstring 01100001, and the character "b" is encoded as 01100010. We can express a message that consists of several characters by concatenating the bitstrings that correspond to each character

$$\text{"baba"} \quad \Leftrightarrow \quad 01100010\ 01100001\ 01100010\ 01100001.$$

For example, if the secret message $\vec{m}$ is the text "beer@18h", we'll represent it as a sequence of eight chunks of eight bits each, making for a bitstring of total length 64:

$$\vec{m} = \underbrace{01100010}_{\text{b}}\ \underbrace{01100101}_{\text{e}}\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \underbrace{01101000}_{\text{h}} \quad \in \quad \{0,1\}^{64}.$$

Representing the text "beer@18h" as a 64-dimensional binary vector, allows us to manipulate it using linear algebra operations. This may sound like a little thing, a convenience at best, but it's actually a big deal. Linear algebra is powerful stuff, and if we can manipulate information using the tools of linear algebra, then we can do powerful things with information. In this section we'll describe how basic vector operations on bitstrings can be used for cryptography. In the next section we'll learn about *error correcting codes* which use matrix operations on binary information vectors to protect them from noise. Put your seatbelt on and get ready.

The XOR operation

The XOR operation, usually denoted $\oplus$, corresponds to addition in the finite field $\mathbb{F}_2$:

$$c = a \oplus b \quad \Leftrightarrow \quad c = (a + b) \mod 2.$$

Table 8.1 shows the results of the XOR operation on all possible combinations of inputs.

$\oplus$	0	1
0	0	1
1	1	0

Table 8.1: The XOR operation $\oplus$ applied to all possible binary input.

The XOR operation acting on two bitstrings of the same length is the XOR operation applied to each of their elements:

$$\vec{c} = \vec{a} \oplus \vec{b} \quad \Leftrightarrow \quad c_i = (a_i + b_i) \mod 2.$$

The XOR operation isn't anything new—it's just the normal vector addition operation for vectors with coefficients in the finite field $\mathbb{F}_2$.

Intuitively, you can think of XOR as a "conditional toggle" operation. Computing the XOR of a bit $a \in \mathbb{F}_2$ with 0 leaves the bit unchanged, while computing the XOR with 1 toggles it, changing 0 to 1 or 1 to 0. To illustrate this "toggle" effect, consider the bitstring

$\vec{v} = 0010$ and the results of XOR-ing it with the all-zeros bitstring 0000, the all-ones bitstring 1111, and a random-looking bitstring 0101:

$$0010 \oplus 0000 = 0010, \quad 0010 \oplus 1111 = 1101, \quad 0010 \oplus 0101 = 0111.$$

In the first case none of the bits are flipped, in the second case all the bits are flipped, and in the third case only the second and fourth bits are flipped.

An important property of the XOR operation is that it is its own inverse: $\vec{z} \oplus \vec{z} = \vec{0}$, for all $\vec{z}$. In particular, if XOR-ing the bitstring $\vec{x}$ with some bitstring $\vec{k}$ produces $\vec{y}$, then XOR-ing $\vec{y}$ with $\vec{k}$ recovers the original vector $\vec{x}$:

$$\text{if } \ \vec{x} \oplus \vec{k} = \vec{y}, \quad \text{ then } \ \vec{y} \oplus \vec{k} = \vec{x}.$$

This statement follows from the self-inverse property of the XOR operation and the associative law for vector addition:

$$\begin{aligned} \vec{y} \oplus \vec{k} &= (\vec{x} \oplus \vec{k}) \oplus \vec{k} \\ &= \vec{x} \oplus (\vec{k} \oplus \vec{k}) \\ &= \vec{x} \oplus \vec{0} \\ &= \vec{x}. \end{aligned}$$

The self-inverse property of the XOR operation is the basis for the one-time pad encryption and decryption operations.

One-time pad crypto system

The one-time pad encryption scheme is based on computing the XOR of the message $\vec{m}$ and the secret key $\vec{k}$, which is of the same length as the message. The security of the protocol depends crucially on how the secret key $\vec{k}$ is generated, and how it is used.

- The key must be kept secret. Only Alice and Bob know $\vec{k}$.
- The key has to be random: the value of each bit of the key $k_i \in \vec{k}$ is random: $k_i = 1$ with probability 50%, and $k_i = 0$ with probability 50%.
- Alice and Bob must *never reuse* any of the secret key.

One-time pad encryption

To obtain the ciphertext $\vec{c}$ for the message $\vec{m}$, Alice computes the XOR of $\vec{m}$ and the secret key $\vec{k}$:

$$\text{Enc}(\vec{m}, \vec{k}) \equiv \vec{m} \oplus \vec{k} = \vec{c}.$$

One-time pad decryption

Upon receiving the ciphertext $\vec{c}$, Bob decrypts it by computing the XOR with the secret key:

$$\text{Dec}(\vec{c}, \vec{k}) \equiv \vec{c} \oplus \vec{k} = \vec{m}.$$

From the self-inverse property of the XOR operation, we know XORing any bitstring with the secret key $\vec{k}$ twice acts like the identity operation: $\vec{m} \oplus \vec{k} \oplus \vec{k} = \vec{m}$.

Discussion

Observe that using this encryption method to send an n-bit message requires n bits of secret key. Since bits of the secret key cannot be reused, we say that sending n bits securely "consumes" n bits of secret key. This notion of secret key as a "consumable resource" is an important general theme in cryptography. Indeed, coming up with encryption and decryption functions is considered the easy part of cryptography, and the hard parts of cryptography are *key management* and *key distribution*.

Consider a practical use of the one-time pad encryption scheme. Sending multiple messages m_1, m_2, ... will require multiple secret keys $\vec{k}_1, \vec{k}_2$, ... that Alice would use when she wants to communicate with Bob. Imagine a pad of pages, each page of the pad containing one secret key $\vec{k}_i$. Since keys cannot be reused, Alice will have to keep flipping through the pad, using the key from each page only once. This is where the name "one-time pad" comes from.

One-time pad security

Security definitions in cryptography are based on different assumptions about the powers of the eavesdropper Eve. We need to formally define what *secure* means, before we can evaluate the security of any crypto system. Modern cryptography is a subfield of mathematics, so we shouldn't be surprised if the definition of security is stated in mathematical terms.

Definition: Indistinguishability under chosen-plaintext attack (IND-CPA) A cryptosystem is considered secure in terms of indistinguishability if no Eve can distinguish the ciphertexts of two messages $\vec{m}_a$ and $\vec{m}_b$ chosen by the eavesdropper, with probability greater than guessing randomly.

This is one of the strongest type of security standards we can expect from a crypto system. It is also the simplest to understand. This

definition of security can be understood as a game played between Alice and Eve. Suppose Eve can force Alice to send only one of two possible messages $\vec{m}_a$ or $\vec{m}_b$. Every time Eve sees a ciphertext $\vec{c}$, she just has to guess whether Alice sent $\vec{m}_a$ or $\vec{m}_b$. We want to compare an Eve that has access to $\vec{c}$, with a "control Eve" that only has access to a random string $\vec{r}$ completely uncorrelated with the message or the ciphertext. A crypto system is secure according to the *indistinguishability under chosen-plaintext* security definition, if the Eve that has access to $\vec{c}$ is no better than randomly guessing what $\vec{m}$ is (the best that "control Eve" could do).

We won't go into the formal details of the math, but we need to specify exactly what we mean by "with probability greater than guessing randomly." Control Eve has a completely random string $\vec{r}$, which contains no information about the message $\vec{m}$, so control Eve must guess randomly and her probability of success is $\frac{1}{2}$. An Eve that can distinguishing the ciphertext $\vec{c}_a \equiv \text{Enc}(\vec{m}_a)$ from the ciphertext $\vec{c}_b \equiv \text{Enc}(\vec{m}_b)$ with a probability significantly greater than $\frac{1}{2}$ is considered to have an "advantage" in distinguishing the ciphertext. Any such scheme is not considered secure in terms of IND-CPA. Intuitively, the IND-CPA definition of security captures the notion that Eve should learn no information about the message $\vec{m}$ after seeing its ciphertext $\vec{c}$.

Sketch of security proof The one-time pad encryption system is secure according to IND-CPA because of the the assumption we make about the shared secret key $\vec{k}$, namely that it is generated from the random binary distribution. Each bit $k_i \in \vec{k}$ is 1 with probability 50%, and 0 with probability 50%.

Eve knows the plaintext of the two messages m_a and m_b, but she can't tell which from the ciphertext $\vec{c}$ because she can't distinguish between these two equally likely alternative scenarios. In Scenario 1 Alice sent $\vec{m}_a$, the secret key is $\vec{k}_a$, where $\vec{c} = \vec{m}_a \oplus \vec{k}_a$. In Scenario 2 Alice sent $\vec{m}_b$, the secret key is $\vec{k}_b$, where $\vec{c} = \vec{m}_b \oplus \vec{k}_b$. The XOR operation "mixes" the randomness of $\vec{m}$ with the randomness in the secret key $\vec{k}$, so trying to distinguish whether $\vec{m}_a$ or $\vec{m}_b$ was sent is just as difficult as distinguishing $\vec{k}_a$ and $\vec{k}_b$. Since $\vec{k}$ is completely random, Eve is forced to guess randomly. Thus the probability of determining the correct message is no better than guessing, key pairs which is precisely the requirement for the definition of security.

The randomness of the shared secret key is crucial to the security of the one-time pad encryption scheme. In general we can think of *shared randomness* (shared secret key) as a communication resource that allows for mathematically-secure private communication between two parties. But what if Alice and Bob don't have access to shared

randomness (some shared secret)? In the next section we'll introduce a different type of crypto system which doesn't depend on a shared secret between Alice and Bob.

Public key cryptography

Assume Alice and Bob are dissidents in two neighbouring countries who want to communicate with each other, but they can't trust the network that connects them. If the dissidents cannot meet in person, obtaining a shared secret key to use for encryption will be difficult. They can't send the secret key $\vec{k}$ over the network because the eavesdropper will see it, and thus be able to decrypt all subsequent encrypted communications between the dissidents. This is a major limitation of all *symmetric-key* crypto systems in which the same secret is used to encrypt and decrypt messages.

Do not despair dear readers, the system hasn't won yet: the dissidents can use *public-key* cryptography techniques, to share a secret key over the untrusted network, in plain view of all state and non-state sponsored eavesdroppers. The security of public-key crypto systems comes from some clever math operations (in a finite field), and the computational difficulty of "reversing" these mathematical operations for very large numbers. For example, the security of the *Rivest–Shamir–Adleman* (RSA) crypto system depends on the difficulty of factoring integers. It is easy to compute the product of two integers d and e, but given only the product de it is computationally difficult to find the factors e and d. For large prime numbers e and d, even the letter agencies will have a difficult time finding the factor of de. Citizens using math to defend against corporations spying and the police state—this is bound to be a central theme in the 21$^{\text{st}}$ century.

Definitions

In a public-key crypto system the secret key is actually a pair of keys $\vec{k} \equiv \{e_{\vec{k}}, d_{\vec{k}}\}$, where $e_{\vec{k}}$ is the public encryption key and $d_{\vec{k}}$ is the private decryption key. The same function Enc is used for encryption and decryption, but with different parts of the key.

To use public-key cryptography, each communicating party must generate their own public-private key pairs. We'll focus on Alice, but assume Bob performs analogous steps. Alice generates a public-private key pair $\{e_{\vec{k}}, d_{\vec{k}}\}$, then she must shares the public part of the key with Bob. Note the public key can be shared in the open, and it's not a problem if Eve intercepts it—this is why it's called a public key—everyone is allowed to know it.

Encryption

Bob will use Alice's public encryption key whenever he wants to send Alice a secret message. To encrypt message $\vec{m}$, Bob will use the function Enc and Alice's public encryption key $e_{\vec{k}}$ as follows:

$$\vec{c} = \text{Enc}(\vec{m}, e_{\vec{k}}).$$

When Alice receives the ciphertext $\vec{c}$, she'll use her private key $d_{\vec{k}}$ (that only she knows) to decrypt the message:

$$\vec{m} = \text{Enc}(\vec{c}, d_{\vec{k}}).$$

Observe that public-key crypto systems are inherently many-to-one: anyone who knows Alice's public key $e_{\vec{k}}$ can create encrypted messages that only she can decode.

Digital signatures

Alice can also use her public-private key pair to broadcast one-to-many *authenticated* statements $\vec{s}$, meaning receivers can be sure the statements they receive were sent by Alice. The math is the same: we just use the keys in the opposite order. Alice encrypts the statement $\vec{s}$ to produce a ciphertext

$$\vec{c} = \text{Enc}(\vec{s}, d_{\vec{k}}),$$

then posts the encrypted post $\vec{c}$ to her blog or on a public forum. Everyone who know Alice's public key $e_{\vec{k}}$ can decrypt the post $\vec{c}$ to obtain the statement $\vec{s}$:

$$\vec{s} = \text{Enc}(\vec{c}, e_{\vec{k}}).$$

The interesting property here is that we can be sure the statement $\vec{s}$ was sent by Alice, since only she controls the private key $d_{\vec{k}}$. This digital signature scheme makes it very difficult for any third parties to impersonate Alice since they don't know Alice's private key $d_{\vec{k}}$. This is the principle behind *digital signatures* used in the delivery of software updates.

We don't have the space in this section to go any deeper into public-key cryptography, but we'll illustrate the main ideas through some practical examples.

Example 1: encrypting emails using GPG

By default all email communications happens in plaintext. When you send and email, this email will be stored in plaintext at your email

hosting provider, travel over the Internet, and then be stored in plaintext at the receiver's hosting provider. There is basically zero privacy when using email. That's why you should never send passwords or other credentials by email. Email messages are also subject to impersonation. Indeed, it's fairly easy to create email messages that appear to originate from someone else. Spammers love this fact because they know you're much more likely to open emails sent by your friends.

The GNU Privacy Guard (GPG) is a suite of cryptographic software specifically tailored for email communication. Using the command line tool `gpg` or an appropriate plugin for your favourite email client, it's possible to achieve private and authenticated communication. Using GPG is like a giant middle finger raised toward the letter agencies who want to snoop in on your communications.

We'll now discuss the basic steps to get started with using GPG. The first step is to install gpg tools for your operating system from the GPG website `https://www.gnupg.org/download/`. The next step is to generate the private-public key pairs that you'll use to encrypt and sign your emails. This is accomplished using the following command:

```
$ gpg --gen-key
```

When prompted, choose the default option of RSA and RSA, key size of 4096 (make the bastards work!), and validity of 0 so the key won't ever expire. Enter your full name when the wizard prompts you for "real name" and your email address at the next prompt. It's important to choose a passphrase for your gpg keys, otherwise anyone who gets a hold of your computer could use these keys to impersonate you.

Once all this information is entered, the program will run for some time and generate the keys. When the program is done, it will print a bunch of information on the screen. You'll want to note the key id, which is an eight-character long ASCII string that appears in place of the question marks in "`pub 4096R/????????`."

The next step is to upload your public key the a key server. This will allow people who want to communicate with you securely to get a hold of your public keys, by searching for you based on your name or your email. Use the following command to upload your public key to the `pgp.mit.edu`, a very popular key server operated by MIT:

```
$ gpg --send-keys --keyserver pgp.mit.edu ????????
```

Be sure to replace `????????` with the hexadecimal key id you noted when you generated your keys. With this step, you're done setting up and publishing your keys.

If you want to communicate securely with a friend, you'll first need to obtain their public key from the keyserver. Using this command to search the keyserver `pgp.mit.edu` based on your friend's name or email:

```
$ gpg --keyserver pgp.mit.edu --search-keys <friend's email>
```

If you find a key for your friend, you can follow the steps in the wizard to import your friend's public key into your keychain. I've got my key posted there (my key id is `C0D34F08`, with key fingerprint `8CBB AA5B CDA6 ...`), so if none of your friends are using PGP yet, you can add me by searching for `ivan savov`. Run `gpg --list-keys` to see all the keys you have on your keychain.

At this point we're done with all the preliminaries. Let's test the `gpg` setup by sending an encrypted message to `urfriend@host.com`. Use you favourite editor to write your message then save it as a text file called `msg.txt`. Next run the following command:

```
$ gpg --encrypt -r urfriend@host.com --sign --armor msg.txt
```

This will encrypt the contents of the `msg.txt` using the recipient's (`-r`) public key, sign the message using your private key, and express the output as an ascii-armoured plain text file `msg.txt.asc` in the same directory. Copy-paste the contents of this file (including the header) into a regular email and send it to your friend.

The procedure for verifying the signature and decrypting an email you receive from your friend is much simpler. Copy-paste the contents of the entire encrypted email into a temporary file `newemail.txt.asc` then you run the command

```
$ gpg newemail.txt.asc
```

This will confirm the email was really sent by your friend and decrypt the contents of the message creating a new file called `newemail.txt`. There you have it: secure email comms that letter agencies can't read.

The steps described above involve calling manually the `gpg` command from the command line, but in practice GPG is integrated into most desktop email software. For example, installing `GPGTools` for Mac integrates GPG with the Mail application, making sending encrypted emails as simple as clicking a button. GPG is now a well established technology, so it is likely your favourite desktop email program has a GPG plugin that you can start using today.

Example 2: ssh keys for remote logins

Connecting to a remote host can be achieved using the secure shell protocol. Running the command `ssh user@remotehost.com` will attempt to login as `user` on the server `remotehost.com`, asking your to provide a password. Passwords are the weakest form of authentication. Given enough effort, it's always possible for an attacker to guess your password. Also, people tend to often use the same password on

different services, which means if one service is compromised then all services will be compromised. In this section we'll describe a more secure approach for ssh logins based on public key authentication.

Setting up key-based authentication for `ssh` requires steps to be performed on your machine and on the remote server. Commands prefixed with `laptop$` should be typed on your local machine, while commands that start with `remotehost$` are to be executed on the remote server. Note these commands assume a UNIX environment. If you're using a Windows machine, I recommend you install `cygwin` to get the same functionality.

The first step is to generate an `ssh` private-public key pair in the directory called `.ssh` inside your home directory:

```
laptop$ cd ~                              # go to you home dir ~
laptop$ mkdir .ssh                        # create a .ssh subdir
laptop$ ssh-keygen -t rsa -b 4096         # generate an RSA key pair
```

Accept the defaults for the questions you're asked, and be sure to set a password to protect your `ssh` private key. A new public key pair will be created in the directory `~/.ssh/`. The private key is contained in the file `~/.ssh/id_rsa` and the corresponding public key is in the file `~/.ssh/id_rsa.pub`. You can confirm these files exist by running the command `ls .ssh`. You can ensure these files have restrictive access permissions by issuing the command:

```
laptop$ chmod -R go-rwx .ssh
```

The above command has the effect of recursively (`-R`) changing permissions users in the same group (`g`) as you and other (`o`) users, by removing (`-`), read (`r`), write (`w`), and execute (`x`) permissions. Basically nobody should be allowed to touch these files.

Another useful command to know is how to add the private key to a "key chain," which is a temporary store of credentials tied to your current login session on your laptop:

```
laptop$ ssh-add -K ~/.ssh/id_rsa    # add priv. key to keychain
```

This will prompt you to enter the password you used when generating the `ssh` key pair. Now that the private key is on your keychain, you won't be prompted for a password for the next little while.

Next, we want to copy the public key `~/.ssh/id_rsa.pub` from your laptop to the remote server. We can do this using the command `scp` as follows:

```
laptop$ scp id_rsa.pub user@remotehost:~/key_from_laptop.pub
```

You'll need to substitute `user` and `remotehost` with the actual username and hostname (or IP address) of the remote server for which you want to setup ssh-key login.

The final step is to place the the public key information in a special file called `~/.ssh/authorized_keys` on the remote host. We can do this using the following commands:

```
laptop$ ssh user@remotehost
remotehost$ cd ~
remotehost$ mkdir .ssh
remotehost$ cat key_from_laptop.pub >> .ssh/authorized_keys
remotehost$ chmod -R go-rwx .ssh
```

If you inspect the resulting file you'll see it now contains a copy of the public key from your laptop. If you now logout from the remote host and try to reconnect to it using `ssh user@remotehost.com`, you'll get logged in automatically without the need for password.

Once you've setup login for your server using `ssh` keys, it would be a good idea to completely disable password logins. In general, this is the best approach to follow and many hosting services already use this approach.

Discussion

We'll conclude this section with important advice for developers who want to use cryptography in their programs. The main thing to keep in mind is not to try to roll your own crypto functions but to use established libraries. There are many ways that crypto systems can be attacked, and people have thought about defending against these attacks. Libraries are good for you. Use them. Don't be a cowboy programmer.

Links

[Visual cryptography]
`http://www.datagenetics.com/blog/november32013/`

[Crypto advice for developers]
`http://daemonology.net/blog/2009-06-11-cryptographic-right-answers.html`

[Public-key cryptography general concepts]
`https://en.wikipedia.org/wiki/Public-key_cryptography`

8.10 Error correcting codes

The raw information carrying capacity of a DVD is about 5.64GB, which is about 20% more than the 4.7GB of data that your computer will let you write to it. Why this overhead? Are DVD manufacturers trying to cheat you? No, they're actually looking out for you; the

extra space is required for the *error correcting code* that is applied to your data before writing it to the disk. Without the error correcting code, even the tiniest scratch on the surface of the disk would make the disk unreadable and destroy your precious data. In this section we'll learn how error correcting codes work.

Error correcting codes play an essential part in the storage, the transmission, and the processing of digital information. Even the slightest change to a computer program will make it crash—computer programs simply don't like it when you fiddle with their bits. Crashing programs were the norm back in the 1940s as illustrated by this quote:

> "Two weekends in a row I came in and found that all my stuff had been dumped and nothing was done. I was really annoyed because I wanted those answers and two weekends had been lost. And so I said, Dammit, **if the machine can detect an error, why can't it locate the position of the error and correct it?**"
>
> —Richard Hamming

Richard Hamming, who was a researcher at Bell in the 1940s, ran into the problem of digital data corruption and decided to do something to fix it. He figured out a clever way to encode k bits of information into n bits of storage, such that it is possible to recover the information even if some errors occur on the storage medium. An *error correcting code* is a mathematical strategy for defending against erasures and errors. Hamming's invention of error correcting codes was a prerequisite for the modern age of computing: reliable computation is much more useful than unreliable computation.

Definitions

An *error correcting code* is a prescription for encoding *binary* information. Recall bits are the element of the finite field with two elements $\mathbb{F}_2 = \{0, 1\}$. A bitstring of length n is an n-dimensional vector of bits $\vec{v} \in \{0,1\}^n$. For example, 0010 is a bitstring of length 4.

We use several parameters to characterize *error correcting codes*:

- k: the size of the messages for the code
- $\vec{x}_i \in \{0,1\}^k$: a *message*. Any bitstring of length k is a valid message.
- n: the size of the codewords in the code
- $\vec{c}_i \in \{0,1\}^n$: the *codeword* that corresponds to message $\vec{x}_i$
- A *code* consists of 2^k codewords $\{\vec{c}_1, \vec{c}_2, \ldots\}$, one for each of the possible messages $\{\vec{x}_1, \vec{x}_2, \ldots\}$.
- $d(\vec{c}_i, \vec{c}_j)$: the *Hamming distance* between codewords $\vec{c}_i$ and $\vec{c}_j$

- An (n, k, d) code is a procedure for encoding messages into codewords $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$ which guarantees the *minimum distance* between any two codewords is at least d.

The *Hamming distance* between two bitstrings $\vec{x}, \vec{y} \in \{0,1\}^n$ counts the number of bits where the two bitstrings differ:

$$d(\vec{x}, \vec{y}) \equiv \sum_{i=1}^{n} \delta(x_i, y_i), \quad \text{where } \delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i, \\ 1 & \text{if } x_i \neq y_i. \end{cases}$$

Intuitively, the Hamming distance between two bitstrings measures the minimum number of substitutions required to transform one bitstring into the other. For example, the the Hamming distance between codewords $\vec{c}_1 = 0010$ and $\vec{c}_2 = 0101$ is $d(\vec{c}_1, \vec{c}_2) = 3$ because it takes three substitutions (also called *bit flips*) to convert $\vec{c}_1$ to $\vec{c}_2$ or vice versa.

An (n, k, d) code is defined by an function $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$ that encoding messages $\vec{x}_i \in \{0,1\}^k$ into codewords $\vec{c}_i \in \{0,1\}^n$. Usually the encoding procedure Enc is paired with a decoding procedure $\text{Dec} : \{0,1\}^n \to \{0,1\}^k$ for recovering messages from (possibly corrupted) codewords.

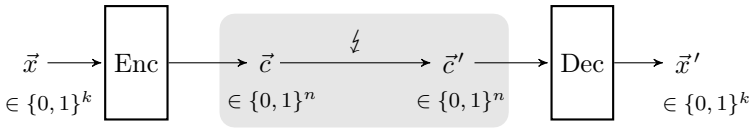


Figure 8.14: An error correcting scheme using the encoding function Enc and the decoding function Dec to protect against the effect of noise (denoted $\lightning$). Each message $\vec{x}$ is encoded into a codeword $\vec{c}$. The codeword $\vec{c}$ is transmitted through a *noisy channel* that can corrupt the codeword transforming it to another bitstring $\vec{c}'$. The decoding function Dec will look for a valid codeword $\vec{c}$ that is close in Hamming distance to $\vec{c}'$. If the end-to-end protocol is successful, the decoded message will match the transmitted message $\vec{x}' = \vec{x}$, despite the noise ($\lightning$).

Linear codes

A code is *linear* if its encoding function Enc is a linear transformation:

$$\text{Enc}(\vec{x}_i + \vec{x}_j) = \text{Enc}(\vec{x}_i) + \text{Enc}(\vec{x}_j), \text{ for all messages } \vec{x}_i, \vec{x}_j.$$

A linear code that encodes k-bit messages into n-bit codewords with minimum inter-codeword distance d is denoted $[n, k, d]$. We use square brackets to enclose the code parameters to indicate the code has a

linear structure. Linear codes are interesting because their encoding function Enc can be implemented as a matrix multiplication.

- $G \in \mathbb{F}_2^{k \times n}$: the *generating matrix* of the code. Each codeword $\vec{c}_i$ is produced by multiplying the message $\vec{x}_i$ by G from the right:

$$\text{Enc}(\vec{x}_i) = \vec{c}_i = \vec{x}_i G.$$

- $\mathcal{R}(G)$: the row space of the generator matrix is called the *code space*. We say a codeword $\vec{c}$ is valid if $\vec{c} \in \mathcal{R}(G)$, which means there exists some message $\vec{x} \in \{0,1\}^k$ such that $\vec{x}G = \vec{c}$.
- $H \in \mathbb{F}_2^{(n-k) \times n}$: the *parity check matrix* of the code. The *syndrome* vector $\vec{s}$ of any bitstring $\vec{c}'$ is obtained by multiplying $\vec{c}'^{\mathsf{T}}$ by H from the left:

$$\vec{s} = H\vec{c}'^{\mathsf{T}}.$$

  If $\vec{c}'$ is a valid codeword (no error occurred) then $\vec{s} = \vec{0}$. If $\vec{s} \neq \vec{0}$, we know an error has occurred. The syndrome information helps us to correct the error.

We can understand linear codes in terms of the input and output spaces of the encoding function $\text{Enc}(\vec{x}) \equiv \vec{x}G$. Left multiplication of G by a k-dimensional row vector produces a linear combination of the rows of G. Thus, the set of all possible codewords (called the *code space*) corresponds to the row space of G.

Every vector in the null space of G is orthogonal to every codeword $\vec{c}_i$. We can construct a parity check matrix H by choosing any basis for the null space for G. We call H the orthogonal complement of G, which means $\mathcal{N}(G) = \mathcal{R}(H)$. Alternately, we can say the space of n-dimensional bitstrings decomposes into orthogonal subspaces of valid and invalid codewords: $\mathbb{F}_2^n = \mathcal{R}(G) \oplus \mathcal{R}(H)$. We know $H\vec{c}^{\mathsf{T}} = \vec{0}$ for all valid codeword $\vec{c}$. Furthermore, the *syndrome* obtained by multiplying an invalid codeword $\vec{c}'$ with the parity check matrix $\vec{s} = H\vec{c}'^{\mathsf{T}}$ can help us characterize the error that occurred, and correct it.

Coding theory

The general idea behind error correcting codes is to choose the 2^k codewords so they are placed far apart from each other in the space $\{0,1\}^n$. If a code has minimum distance between codewords $d \geq 2$, then this code is robust to one-bit errors. To understand why, imagine a bubble of radius 1 (in Hamming distance) around each codeword. When a one-bit error occurs, a codeword will be displaced from its position, but it will remain within the bubble of radius one. In other

words, if a one-bit error occurs, we can still find the correct codeword by looking for the closest valid codeword. See Figure 8.15 for an illustration of a set of codewords that are $d > 2$ distance apart. Any bitstring that falls within one of the bubbles will be decoded as the codeword at the centre of the bubble. We cannot guarantee this decoding procedure will succeed if more than one errors occur.

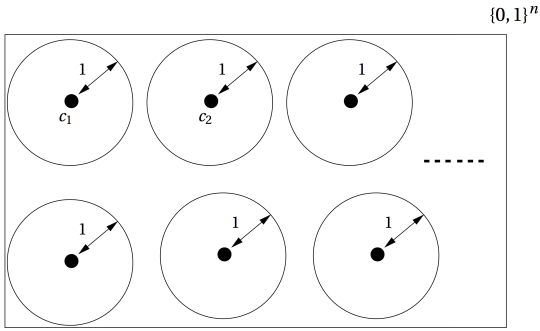


Figure 8.15: The rectangular region represents the space of binary strings (bitstrings) of length n. Each codeword c_i is denoted with a black dot. A "bubble" of Hamming distance one around each codeword is shown. Observe the distance between any two codewords is greater than two ($d > 2$). By Observation 1, we know this code can correct any one-bit error ($\lfloor \frac{d}{2} \rfloor \geq 1$).

Observation 1 An (n, k, d)-code can correct up to $\lfloor \frac{d}{2} \rfloor$ errors.

The notation $\lfloor x \rfloor$ describes the *floor* function, which computes the closest integer value smaller than x. For example, $\lfloor 2 \rfloor = 2$ and $\lfloor \frac{3}{2} \rfloor = \lfloor 1.5 \rfloor = 1$. We can visualize Observation 1 using Figure 8.15 by imagining the radius of each bubble is $\lfloor \frac{d}{2} \rfloor$ instead of 1.

Repetition code

The simplest possible error correcting code is the *repetition code*, which protects the information by recoding multiple copies of each message bit. For instance, we can construct a $[3, 1, 3]$ code by repeating each message bit three times. The encoding procedure Enc is defined as follows:

$$\text{Enc}(0) = 000 \equiv \vec{c}_0, \qquad \text{Enc}(1) = 111 \equiv \vec{c}_1.$$

Three bit flips are required to change the codeword $\vec{c}_0$ into the codeword $\vec{c}_1$, and vice versa. The Hamming distance between the codewords of this repetition code is $d = 3$.

Encoding a string of messages $x_1x_2x_3 = 010$ will result in a string of codewords 000111000. We can use the "majority vote" decoding strategy using the following decoding function Dec defined by

$$\text{Dec}(000) = 0,\ \text{Dec}(100) = 0,\ \text{Dec}(010) = 0,\ \text{Dec}(001) = 0,$$
$$\text{Dec}(111) = 1,\ \text{Dec}(011) = 1,\ \text{Dec}(101) = 1,\ \text{Dec}(110) = 1.$$

Observe that any one-bit error is corrected. For example, the message $x = 0$ will be encoded as the codeword $\vec{c} = 000$. If an error occurs on the first bit during transmission, the received codeword will be $\vec{c}' = 100$, and majority-vote decoding will correctly output $x = 0$. Since $d > 2$ for this repetition code, it can correct all one-bit errors.

The Hamming code

The $[7, 4, 3]$ *Hamming code* is a linear code that encodes four-bit messages into seven-bit codewords with minimum Hamming distance of $d = 3$ between any two codewords. The generator matrix for the Hamming code is

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Note that other possibilities for the matrix G exist. Any permutation of the columns and rows of the matrix will be a generator matrix for a $[7, 4, 3]$ Hamming code. We have chosen this particular G because of the useful structure in its parity check matrix H, which we'll discuss shortly.

Encoding

We'll now look at how the generating matrix is used to encode four-bit messages into seven-bit codewords. Recall that all arithmetic operations are performed in the finite field $\mathbb{F}_2$. The message $(0, 0, 0, 1)$ will be encoded as the codeword

$$(0, 0, 0, 1)G = (0, 0, 0, 0, 1, 1, 1),$$

similarly $(0, 0, 1, 0)$ will be encoded into $(0, 0, 1, 0)G = (0, 0, 1, 1, 0, 0, 1)$. Now consider the message $(0, 0, 1, 1)$, which is a linear combination of the messages $(0, 0, 1, 0)$ and $(0, 0, 0, 1)$. To obtain the codeword for this message we can multiply it with G as usual to find $(0, 0, 1, 1)G = (0, 0, 1, 1, 1, 1, 0)$. Another approach would be to use the linearity of the code and add the codewords for the messages $(0, 0, 1, 0)$ and $(0, 0, 0, 1)$: $(0, 0, 1, 1, 0, 0, 1) + (0, 0, 0, 0, 1, 1, 1) = (0, 0, 1, 1, 1, 1, 0)$.

Decoding with error correction

The minimum distance for this Hamming code is $d = 3$, which means it can correct one-bit errors. In this section we'll look at some examples of bit fiip errors that can occur, and discuss the decoding procedure we can follow to extract the message even from a corrupted codeword $\vec{c}'$.

The *parity check matrix* for the $[7, 4, 3]$ Hamming code is

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

This matrix is the orthogonal complement of the generating matrix G. Every valid codeword $\vec{c}$ is in the row space of G, since $\vec{c} = \vec{x}G$ for some message $\vec{x}$. Since the rows of H are orthogonal to $\mathcal{R}(G)$, the product of H with any valid codeword will be zero: $H\vec{c}^{\mathsf{T}} = 0$.

On the other hand, if the codeword $\vec{c}'$ contains an error, then multiplying it with H will produce a nonzero *syndrome* vector $\vec{s}$:

$$H\vec{c}'^{\mathsf{T}} = \vec{s} \neq 0.$$

The decoding procedure Dec can use the information in the syndrome vector $\vec{s}$ to correct the error. In general, the decoding function could be a complex procedure that involving $\vec{s}$ and $vecc'$. In the case of the the Hamming code, the decoding procedure is very simple because the syndrome vector $\vec{s} \in \{0, 1\}^3$ contains the binary representation of the location where the bit-flip error occurred. Let's look at how this works through some examples.

Suppose we sent the message $\vec{x} = (0, 0, 1, 1)$ encoded as the codeword $\vec{c} = (0, 0, 1, 1, 1, 1, 0)$. If an error on the last bit occurs in transit (bit one counting from the right), the received codeword will be $\vec{c}' = (0, 0, 1, 1, 1, 1, 1)$. Computing the syndrome for $\vec{c}'$ we obtain

$$\vec{s} = H\vec{c}'^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The syndrome vector $\vec{s} = (0, 0, 1)$ corresponds to the binary string 001, which is the number one. This is the location where the error occurred—the first bit of the codeword (counting from the right). After correcting the error on the first bit we obtain the correct $\vec{c} =$

$(0,0,1,1,1,1,0)$, which decodes to the message $\vec{x} = (0,0,1,1)$ that was sent.

Let's check the error-correcting ability of the Hamming code with another single-bit error. If a bit-flip error occurs on the fourth bit, the received codeword will be $\vec{c}'' = (0,0,1,0,1,1,0)$. The syndrome for $\vec{c}''$ is $H\vec{c}''^{\mathsf{T}} = (1,0,0)$, which corresponds to the number four when interpreted in binary. Again, we're able to obtain the position of error from the syndrome.

The fact that the syndrome tells us where the error has occurred is not a coincidence, but a consequence of deliberate construction of the matrices G and H of the Hamming code. Let's analyze the possible received codewords $\vec{c}'$, when the transmitted codeword is $\vec{c}'$:

$$\vec{c}' = \begin{cases} \vec{c} & \text{if no error occurs} \\ \vec{c} + \vec{e}_i & \text{if bit-flip error occurs in position } i \end{cases}$$

where $\vec{e}_i$ is vector that contains a single one in position i. Indeed a bit flip in the i^{th} position is the same as adding one modulo two in that position.

In the case when no error occurs the syndrome will be zero $H\vec{c} = \vec{0}$, because H is defined as the orthogonal complement of the code space (the row space of G). In the case of a single error occurs, the syndrome calculation only depends on the error:

$$\vec{s} = H\vec{c}'^{\mathsf{T}} = H(\vec{c} + \vec{e}_i)^{\mathsf{T}} = \cancel{H\vec{c}} + H\vec{e}_i = H\vec{e}_i.$$

If you look carefully at the structure in the parity check matrix H, you'll see its columns contain the binary encoding of the numbers between seven and one. With this clever choice of the matrix H, we're able to obtain a syndrome that tells us the binary representation of where the error has occurred.

Discussion

Throughout this section we referred to "the" [7,4,3] *Hamming code*, but in fact there there is a lot of freedom when defining a Hamming code with these dimensions. For example, we're free to perform any permutation of the columns of the generator matrix G and the parity check matrix H, and the resulting code will have the same properties as the one described above.

The term *Hamming code* actually applies to a whole family of linear codes. For any $r > 2$, there exists a $[2^r - 1, 2^r - r - 1, 3]$ Hamming code, that has similar structure and properties as the *Hamming* $[7,4,3]$ *code* described above.

The ability to "read" the location of the error directly from the syndrome is truly a marvellous mathematical construction, that is

particular to the Hamming code. For other types error correcting codes inferring the error from the syndrome vector $\vec{s}$ may require a more complicated procedure.

Note that Hamming codes all have minimum distance $d = 3$, which means they allow us to correct $\lfloor \frac{3}{2} \rfloor = 1$ bit errors. Hamming codes are therefore not appropriate for use with communication channel on which multi-bit errors are likely to occur. There exist other code families like the *Reed–Muller codes* and *Reed–Solomon codes*, which can be used in more noisy scenarios. For example, Reed–Solomon codes are used by NASA for deep-space communications and for error correction on DVDs.

Error detecting codes

Another approach for dealing with errors is to focus on *detecting* the errors and not trying to correct them. Error detecting codes, like the *parity check code*, are used in scenarios where it is possible to retransmit messages; if the receiver detects a transmission error has occurred, she can ask the sender to retransmit the corrupted message. The receiver will be like "Yo, Sender, I got your message, but its *parity* was *odd*, so I know there was an error and I want you to send that message again." Error detect and retransmit is how Internet protocols work (TCP/IP).

The *parity check code* is a simple example of an error detecting code. The *parity* of a bitstring describes whether the number of 1s in the string is odd or even. The bitstring 0010 has odd parity, while the bitstring 1100 has even parity. We can compute the parity of any bitstring by taking the sum of its bits, the sum being performed in finite field $\mathbb{F}_2$.

A simple $[k+1, k, 2]$ parity check code can be created by appending a single bit p (the parity bit) to end of the every message to indicate the parity of the message bitstring $x_1x_2 \cdots x_k$. We append $p = 1$ if the message has odd parity, and $p = 0$ if the message has even parity. The resultant message-plus-parity-check bitstring $\vec{c} = x_1x_2 \cdots x_kp$ will always have even parity.

If a single bit-flip error occurs during transmission, the received codeword $\vec{c}'$ will have odd parity, which tells us the message data has been affected by noise. More advanced error detecting schemes can detect multiple errors, at the cost of appending more parity-check bits at end of the messages.

Links

[The Hamming distance between bitstrings]
`https://en.wikipedia.org/wiki/Hamming_distance`

[More examples of linear codes on Wikipedia]
`https://en.wikipedia.org/wiki/Linear_code`
`https://en.wikipedia.org/wiki/Hamming_code`
`https://en.wikipedia.org/wiki/Reed-Muller_code`

[Details of error correcting codes used on optical disks]
`http://bat8.inria.fr/~lang/hotlist/cdrom/Documents/tech-summary.html`
`http://usna.edu/Users/math/wdj/_files/documents/reed-sol.htm`
`http://multimediadirector.com/help/technology/cd-rom/cdrom_spec.htm`

Exercises

E8.9 Find the codeword $\vec{c}$ that corresponds to the message $\vec{x} = (1, 0, 1, 1)$ for the $[7, 4, 3]$ Hamming code whose generator matrix G is given on page 371.

E8.10 Construct the encoding matrix for a $[8, 7, 2]$ parity check code.

8.11 Fourier analysis

Way back in the 17^{th} century, Isaac Newton carried out a famous experiment using light beams and glass prisms. He showed that a beam of white light splits into a rainbow of colours upon passing through a prism: starting from red at one end, followed by orange, yellow, green, blue, and finally violet at the other end. This experiment showed that white light is made up of *components* with different colours. Using the language of linear algebra, we can say that white light is a "linear combination" of different colours.

Today we know that the different light colours correspond to electromagnetic waves with different frequencies: red light has frequency around 450 THz, while violet light has frequency around 730 THz. We can therefore say that white light is made up of components with different frequencies. The notion of describing complex phenomena in terms of components with different frequencies is the main idea behind *Fourier analysis*.

Fourier analysis is used to describe sounds, vibrations, electric signals, radio signals, light signals, and many other phenomena. The Fourier transform allows us to represent all these "signals" in terms of components with different frequencies. Indeed, the Fourier transform can be understood as a change-of-basis operation from a time basis to a frequency basis:

$$[\mathbf{v}]_t \qquad \Leftrightarrow \qquad [\mathbf{v}]_f.$$

For example, if $\mathbf{v}$ represents a musical vibration, then $[\mathbf{v}]_t$ corresponds to the vibration as a function of time, while $[\mathbf{v}]_f$ corresponds to the frequency content of the vibration. Depending on the properties of the signal in the time domain and the choice of basis for the frequency domain, different Fourier transformations are possible.

We'll study three different bases for the frequency domain based on orthonormal sets of sinusoidal and complex exponential functions. The *Fourier series* is a representation for continuous periodic functions $f(t) \in \{\mathbb{R} \to \mathbb{R}\}$, that is functions that satisfy $f(T+t) = f(t)$. The Fourier basis used in the Fourier series is the set of sines and cosines of the form $\sin(\frac{2\pi n}{T}t)$ and $\cos(\frac{2\pi n}{T}t)$, which form an orthogonal set. The *Fourier transform* is the continuous version of the Fourier series. Instead of a countable set of frequency components, the frequency representation of the signal is described by a complex-valued continuous function $f(\omega) \in \{\mathbb{R} \to \mathbb{C}\}$. Instead of a continuous time parameter $t \in \mathbb{R}$, certain signals are described in terms of N *samples* from the time signal: $\{f[t]\}_{t\in[0,1,\ldots,N-1]}$. The *Discrete Fourier transform* is a version of the Fourier transform for signals defined at discrete time samples. Table 8.2 shows a summary of these three Fourier-type transformations. The table indicates the class of functions for which the transform applies, the Fourier basis for the transform, and the frequency-domain representation used.

Fourier transformations

Name	Time domain	Fourier basis	Frequency domain
FS	$f(t) \in \{\mathbb{R} \to \mathbb{R}\}$ s.t. $f(t) = f(t+T)$	$1, \{\cos(\frac{2\pi n}{T})\}_{n\in\mathbb{N}_+}, \{\sin(\frac{2\pi n}{T})\}_{n\in\mathbb{N}_+}$	$(a_0, a_1, b_1, \ldots)$
FT	$f(t) \in \{\mathbb{R} \to \mathbb{R}\}$ s.t. $\int_{-\infty}^{\infty} \lvert f(t)\rvert^2\, dt < \infty$	$\{e^{i\omega t}\}_{\omega\in\mathbb{R}}$	$f(\omega) \in \{\mathbb{R} \to \mathbb{C}\}$
DFT	$f[t] \in \{[N] \to \mathbb{R}\}$	$\{e^{\frac{iwt}{N}}\}_{w\in[N]}$	$f[w] \in \{[N] \to \mathbb{C}\}$

Table 8.2: Three important Fourier transformations. Observe the different time domain, Fourier basis, and frequency domains for each transform. The *Fourier series* (**FS**) converts periodic continuous time signals into Fourier coefficients. The *Fourier transform* (**FT**) converts finite-power continuous signal into continuous functions of frequency. The *Discrete Fourier transform* (**DFT**) is the discretized version of the Fourier transform.

Fourier transforms are normally studied by physics and electrical engineering students during their second or third year at university. You, my dear readers, thanks to your understanding of linear algebra,

can have a ten-page sneak-peek preview of the Fourier transformations course right now. In this section, we'll study mathematics behind the Fourier transforms and discuss how they're used in practical signal processing applications. Before we jump into signal processing, let's look at something much simpler—the vibration of a guitar string.

Example 1: Describing the vibrations of a string

Imagine a string of length L that is tied at both ends, like a guitar string. If you pluck this string it will start vibrating. We can describe the displacement of the string from its rest (straight) position as a function $f(x)$, where $x \in [0, L]$. The longest vibration on the string is called the *fundamental*, while the rest of the vibrations are called *overtones*. See Figure 8.16 for an illustration. When you pluck the guitar string you'll hear the fundamental and the overtones as a single tone. The relative prominence of the frequencies varies among instruments.

The way vibrations on a string work is that only certain *modes* of vibration will remain in the long term. Any vibration is possible to begin with, but after a while, the energy in the string "bounces around," reflecting from the two fixed endpoints, and many vibrations cancel out. The only vibrations remaining on the string will be the following sin-like vibrations:

$$\mathbf{e}_1(x) \equiv \sin(\tfrac{\pi}{L}x), \quad \mathbf{e}_2(x) \equiv \sin(\tfrac{2\pi}{L}x), \quad \mathbf{e}_3(x) \equiv \sin(\tfrac{3\pi}{L}x), \quad \ldots \ .$$

Vibrations of the form $\mathbf{e}_n(x) \equiv \sin(\frac{n\pi}{L}x)$ persist because they are stable, which means they satisfy the physics constraints imposed on the string. One constraint on the string is that it's two endpoints are clamped down: $\mathbf{e}_n(0) = 0$ and $\mathbf{e}_n(L) = 0$. Another physics constraint requires that the vibration of the string $f(x)$ must satisfy the equation $f(x) = f(L - x)$. Without going further into the physics of string vibrations,[2] let's just say that a "survival of the fittest" phenomenon occurs, and only the finite set of vibrations $\{\mathbf{e}_n(x)\}_{n \in \mathbb{N}^+}$ will remain on the string in the long run.

The set of functions $\{\mathbf{e}_n(x)\}_{n \in \mathbb{N}^+}$ form a basis for the set of vibrations that satisfy $f(0) = 0$, $f(L) = 0$, and $f(x) = f(L - x)$ on $[0, L]$. Any vibration on the string can be described in terms of a sequence of coefficients $(a_1, a_2, a_3, \ldots)$:

$$f(x) \text{ s.t. } f(x) = f(L - x) \qquad \Leftrightarrow \qquad (a_1, a_2, a_3, a_4, \ldots).$$

The coefficients a_i represent how much of the i^{th} vibrations exists on the string.

[2] You can learn more about the physics of standing waves from this tutorial: `http://www.phy.duke.edu/~rgb/Class/phy51/phy51/node34.html`.

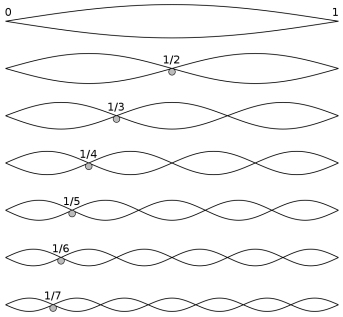


Figure 8.16: Standing waves on a string with length $L = 1$. The longest vibration is called the *fundamental*. Other vibrations are called *overtones*.

Depending on how you pluck the string, the shape of the vibrating string $f(x)$ will be some *superposition* (linear combination) of the vibrations $\mathbf{e}_n(x)$:

$$\begin{aligned} f(x) &= a_1 \sin(\tfrac{\pi}{L}x) + a_2 \sin(\tfrac{2\pi}{L}x) + a_3 \sin(\tfrac{3\pi}{L}x) + \ \cdots \\ &= a_1\,\mathbf{e}_1(x) \quad + a_2\,\mathbf{e}_2(x) \quad + a_3\,\mathbf{e}_3(x) \quad + \ \cdots \end{aligned}$$

That's quite crazy if you think about it: rather than describing the exact shape of the vibrating string $f(x)$, it's sufficient to specify a list of coefficients to describe the same vibration. The coefficient a_1 tells us the prominence of the fundamental vibration, and the other coefficients tells us the prominence of the overtones. Since the laws of physics restricts the possible vibrations to linear combinations of the basis vibrations $\{\mathbf{e}_n(x)\}_{n\in\mathbb{N}^+}$, we can represent any vibration on the string through its sequence of coefficients $(a_1, a_2, a_3, \ldots)$.

The main idea

The above example shows it's possible to describe a complex real-world system like a vibrating guitar string in terms of a succinct description—the coefficients $(a_1, a_2, a_3, \ldots)$. This is general pattern in science that occurs when solving problems stated in terms of differential equations. The solutions can often be expressed as linear combinations of an orthonormal basis of functions. Using this example, we're now in a position to explain the "main idea" behind Fourier transform, connecting it to the concept of inner product space which we studied in Section 7.4.

The fundamental starting point of all Fourier-type reasoning is to define the inner product space for the solution space, and a basis of orthonormal functions for that space. In the case of the sting

vibrations, the space of string vibrations consists of functions $f(x)$ on the interval $[0, L]$ that satisfy $f(0) = 0$, $f(L) = 0$, and $f(x) = f(L-x)$. The inner product operation we'll use for this space is

$$\langle f(x), g(x) \rangle \equiv \int_{x=0}^{x=L} f(x)g(x)\,dx,$$

and the functions $\{\mathbf{e}_n(x)\}_{n\in\mathbb{N}^+}$ form an orthogonal basis for this space. We can verify that $\langle \mathbf{e}_m(x), \mathbf{e}_n(x) \rangle = 0$ for all $m \neq n$. See exercise **E8.11** for a hands-on experience.

Since $\{\mathbf{e}_n(x)\}_{n\in\mathbb{N}^+}$ forms a basis, we can express any function as a linear combination of the basis functions. The complicated-looking Fourier transform formulas that we'll get to shortly can therefore be understood as applications of the general change-of-basis formula.

Change of basis review Let's do a quick review of the change-of-basis formula. You recall the change-of-basis operation, right? Given two orthonormal bases $B = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ and $B' = \{\hat{e}'_1, \hat{e}'_2, \hat{e}'_3\}$, you'd know how to convert a vector $[\vec{v}]_B$ expressed with respect to basis B, into coordinates with respect to basis B'. In case you've forgotten, look back to page 198, where we define the change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ that converts from B coordinates to B' coordinates, and its inverse ${}_{B}[\mathbb{1}]_{B'}$ that converts from B' coordinates to B coordinates:

$$[\vec{v}]_{B'} = {}_{B'}[\mathbb{1}]_B[\vec{v}]_B \qquad \Leftrightarrow \qquad [\vec{v}]_B = {}_{B}[\mathbb{1}]_{B'}[\vec{v}]_{B'}.$$

When the bases are orthonormal, the change-of-basis operation depends only on the inner product between the vectors of the two bases:

$$\vec{v}'_i = \sum_{j=1}^{3} \langle \hat{e}'_i, \hat{e}_j \rangle v_j \qquad \Leftrightarrow \qquad \vec{v}_i = \sum_{j=1}^{3} \langle \hat{e}_i, \hat{e}'_j \rangle v'_j.$$

Orthonormal bases are nice like that—if you know how to compute inner products, you can perform the change of basis.

To apply the general formulas for change of basis to the case of the vibrating string, we must first define the two bases and compute the inner product between them. We can think of the function $f(x) \in \{[0, L] \to \mathbb{R}\}$ as being described in the "default basis" $\mathbf{e}_x$, which is equal to one at x and zero everywhere else. The Fourier basis for the problem consists of the functions $\mathbf{e}_n \equiv \sin(\frac{n\pi}{L}x)$ for $n \in \mathbb{N}^+$. The inner product between $\mathbf{e}_x$ and $\mathbf{e}_n$ is

$$\langle \mathbf{e}_x, \mathbf{e}_n \rangle = \int_0^L \mathbf{e}_x(y) \sin(\tfrac{n\pi}{L}y)\,dy = \sin(\tfrac{n\pi}{L}x),$$

since the function $\mathbf{e}_x(y)$ is equal to zero everywhere except at $y = x$.

We can obtain the change-of-basis transformations by extending notion of matrix-vector appropriately. We use the integral on the interval $[0, L]$ for the change-of-basis to the Fourier coefficients, and the infinite sum over coefficients to change from the space of Fourier coefficients to functions:

$$a_n = \int_0^L \sin(\tfrac{n\pi}{L}x) f(x)\, dx \qquad \Leftrightarrow \qquad f(x) = \sum_{n=1}^{\infty} \sin(\tfrac{n\pi}{L}x) a_n.$$

Compare these formulas with the basic change-of-basis formula between bases B and B' discussed above.

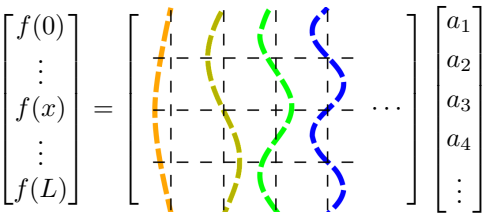


Figure 8.17: Any string vibration $f(x)$ can be represented as coefficients $(a_1, a_2, a_3, a_4, \ldots)$ with respect to the basis of functions $\mathbf{e}_n(x) \equiv \sin(\frac{n\pi}{L}x)$.

Figure 8.17 illustrates right hand side of the change-of-basis formula, which transforms a vibration from the Fourier basis $(a_1, a_2, a_3, a_4, \ldots)$ to a function $f(x) \in \{[0, L] \to \mathbb{R}\}$ in the default basis.

Analysis and synthesis

In the jargon of Fourier transformations, we refer to the change of basis from the default basis to the Fourier basis $({}_f[\mathbb{1}]_x)$ as *Fourier analysis*, and the transformation from the Fourier basis to the the default basis $({}_x[\mathbb{1}]_f)$ as *Fourier synthesis*. This terminology, in addition to sounding extra fancy, gives us some useful intuition about the purpose of the Fourier transform. Given a vibration on the string, we use the Fourier transform to "analyze" the vibration, decomposing it into its constituent vibrations. The *Fourier analysis equations* describe how to calculate each Fourier coefficient.

The opposite direction—starting from the Fourier coefficients and combining them to obtain a vibration default basis is called *synthesis*, in analogy with synthesizer that can generate any sound. The synthesis equations describe how to calculate the values of the vibration as a function of x based on its components in the frequency domain.

For example, you can pick any set of coefficients $(a_1, a_2, a_3, a_4, \ldots)$, form the linear combination $\sum_{n=1}^{N} a_n \sin(\frac{2\pi n}{L}x)$, and check what that

vibration sounds like. This is how synthesizers work: they can reproduce the sound of any instrument by producing the right linear linear combination of vibrations.

Fourier series

When discussing Fourier transformations, it is natural to work with functions of time $f(t)$, which we call *signals*. A signal in the "time basis," is specified by its values $f(t)$ for all times t. This is the "time domain" representation for signals. The *Fourier series* corresponds to the following change-of-basis operation:

$$f(t) \text{ s.t. } f(t) = f(T+t) \qquad \Leftrightarrow \qquad (a_0, a_1, b_1, a_2, b_2, \ldots).$$

The coefficients $(a_0, a_1, b_1, a_2, b_2, \ldots)$ are called the *Fourier coefficients* of $f(t)$. The *Fourier series* applies to all signals $f(t)$, that satisfy the constraint $f(t) = f(t+T)$ for all $t \in [0, T)$. We say such signals are *periodic* with period T.

The basis used for the Fourier series consists of the set of cosine and sine functions with frequencies that are multiples of $\frac{2\pi}{T}$:

$$\left\{\sin\left(\frac{2\pi n}{T}t\right)\right\}_{n\in\mathbb{N}_+} \quad \text{and} \quad \left\{\cos\left(\frac{2\pi n}{T}t\right)\right\}_{n\in\mathbb{N}}.$$

This family of functions forms an orthogonal set with respect to the standard inner product:

$$\langle f(t), g(t)\rangle = \frac{1}{T}\int_{t=0}^{t=T} f(t)g(t)\,dt.$$

Every periodic signal $f(t)$ can be represented as a *Fourier series* of the form:

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(\tfrac{2\pi n}{T}t) \; + \; \sum_{n=1}^{\infty} b_n \sin(\tfrac{2\pi n}{T}t). \qquad \text{(FSS)}$$

The coefficient a_i represent how much of the i^{th} cos-like vibration is contained in $f(t)$, and the the coefficient b_i represents how much of the i^{th} sin-like vibration is contained in $f(t)$. See the illustrated in Figure 8.18. This representation is directly analogous to the analysis we performed for the vibrating string, but this time we use both sines and cosines.

We compute the Fourier coefficients using the following formulas:

$$a_n \equiv \frac{1}{T}\int_0^T f(t)\cos(\tfrac{2\pi n}{T}t)\,dt, \qquad b_n \equiv \frac{1}{T}\int_0^T f(t)\sin(\tfrac{2\pi n}{T}t)\,dt. \quad \text{(FSA)}$$

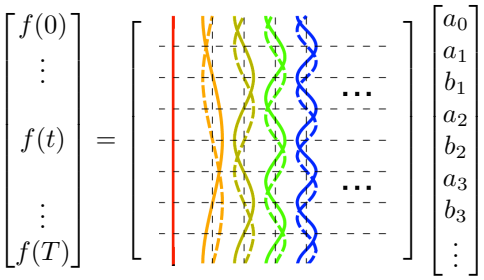


Figure 8.18: A periodic signal $f(t)$ can be represented as a series of Fourier coefficients $(a_0, a_1, b_1, a_2, b_2, a_3, b_3, \ldots)$. The first column corresponds to the constant component $1 = \cos(0)$. The remaining columns correspond to cosines and sines with different frequencies.

These transformations correspond to the standard change of basis for the function $f(t)$ in the time domain to the Fourier basis of cosines and sines.

For the Fourier series representation of a periodic time signal to be exact, we must compute an infinite number of coefficients in the Fourier series $(a_0, a_1, b_1, a_2, b_2, \ldots)$. However, we're often interested in obtaining an approximation to a $f(t)$ using only a finite set of Fourier coefficients:

$$f(t) \approx a_0 + \sum_{n=1}^{N} a_n \cos(\tfrac{2\pi n}{T} t) + \sum_{n=1}^{N} b_n \sin(\tfrac{2\pi n}{T} t).$$

This is called a *Fourier series approximation* since the frequency representation does not contain the components with frequencies $\frac{N+1}{T}$, $\frac{N+2}{T}$, and higher. Nevertheless, these finite-series approximations of signals are used in many practical scenarios; it's much easier to compute a finite number of Fourier coefficients instead of an infinite number.

Example For an example calculation of the Fourier coefficients of the *square wave* signal, see `bit.ly/fourier_series_square_wave` by Joshua Vaughan. Note the square wave analyzed is an *odd* function periodic function, so its coefficients a_n are all zero.

In the next section we'll describe the Fourier transform, which is a continuous-frequency version of the Fourier series.

Fourier transform

The Fourier series representation applies only to periodic functions $f(t)$, but not every function of interest in signal processing is periodic.

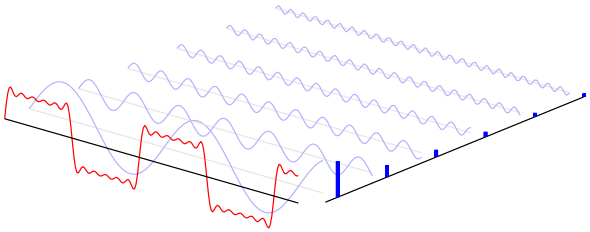


Figure 8.19: The square-wave signal can be approximated by a linear combination of sine functions with different frequencies.

The *Fourier transform* applies to all signals $f(t)$ (periodic or not) that obey the *finite-power* constraint:

$$\int_{t=-\infty}^{t=+\infty} |f(t)|^2 \, dt \;\leq\; \infty.$$

This class of functions includes most signals of practical importance in communication scenarios.

The result of the *Fourier transform* is a complex-valued continuous function in the frequency domain:

$$f(t) \in \{\mathbb{R} \to \mathbb{R}\} \qquad \Leftrightarrow \qquad f(\omega) \in \{\mathbb{R} \to \mathbb{C}\}.$$

The basis used in the Fourier transforms is

$$\mathbf{e}_\omega \;\equiv\; e^{i\omega t}, \text{ for } \omega \in \mathbb{R}.$$

parametrized by the continuous parameters $\omega \in \mathbb{R}$. The set of functions $\{\mathbf{e}_\omega\}$ form an orthogonal basis with respect to the inner product

$$\langle f(t), g(t) \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \overline{f(t)} g(t) \, dt.$$

The change-of-basis from the time domain to the frequency domain is performed using the following integral:

$$f(\omega) \equiv \int \langle \mathbf{e}_\omega, \mathbf{e}_t \rangle f(t) \, dt = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\omega t} f(t) \, dt. \qquad \text{(FTA)}$$

We can understand this formula as an instance of the general change-of-basis formula from the the "default basis" for the time domain $\{\mathbf{e}_t\}_{t\in\mathbb{R}}$, to the basis $\{\mathbf{e}_\omega\}_{\omega\in\mathbb{R}}$. The function $f(\omega)$ is called the *spectrum* or *Fourier transform* of $f(t)$; it tells us all the information about the frequency content of the signal $f(t)$. Note *Fourier transform* can

be used as a verb when referring to the change-of-basis transformation, or as a noun when referring to the function $f(\omega)$, which is the result of the Fourier transform.

The *Inverse Fourier transform* is the change of basis from the frequency domain back to the time domain. Given the frequency representation of a function $f(\omega)$, we can reconstruct the time representation of the signal using the integral:

$$f(t) \equiv \int \langle \mathbf{e}_t, \mathbf{e}_\omega \rangle f(\omega)\, d\omega = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega t} f(\omega)\, d\omega. \qquad \text{(FTS)}$$

Note the similarity between the formulas for the forward and the inverse Fourier transform formulas. Indeed, these are representation of the general change-of-basis formula using the inner product for functions. Compare the Fourier transform change-of-basis formulas and the basic change-of-basis formulas between bases B and B' discussed above (see page 379). The conjugate-symmetric property of the inner product $\langle \mathbf{e}_t, \mathbf{e}_\omega \rangle = \overline{\langle \mathbf{e}_\omega, \mathbf{e}_t \rangle}$ explains the change from $e^{-i\omega t}$ to $e^{i\omega t}$.

Further discussion about the Fourier transform and its many applications is beyond the scope of the current section. If you take a signal processing course (the *best* course in the electrical engineering curriculum), you'll learn all about the Fourier transform.

Discrete Fourier transform

Continuous-time signals are important in sound engineering, radio communications, and other communication scenarios. Signals can also be digitized and represented as discrete samples rather than continuous functions. A continuous-time signal $f(t)$ can be approximated by taking a finite set of N *samples* from the signal. This results of this sampling is a discrete-time signal $f[t] \in \{[N] \to \mathbb{R}\}$, where the shorthand $[N]$ denotes the sequence $[0, 1, \ldots, N-1]$. We'll use square brackets around the function input to distinguish the discrete-time signal $f[t]$ from its continuous-time counterpart $f(t) \in \{\mathbb{R} \to \mathbb{R}\}$.

The *Discrete Fourier transform* converts N samples of a discrete-time signal, into a frequency representation with N frequency components:

$$f[t] \in \mathbb{R}, \text{ for } t \in [N] \qquad \Leftrightarrow \qquad f[w] \in \mathbb{C}, \text{ for } w \in [N].$$

The basis for the frequency domain consists of complex exponentials:

$$\mathbf{e}_w \;\equiv\; e^{i\frac{2\pi w}{N}t}, \text{ for } t \in [N] \text{ and } w \in [N].$$

This set of functions form an orthonormal set with respect to the inner product $\langle f[n], g[n] \rangle \equiv \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \overline{f[n]} g[n]$. You can understand this

inner product and this basis as discrete version of the inner product and basis used for the Fourier transform, hence the name.

To convert from the time domain to the frequency domain, we use the Discrete Fourier transform analysis equation:

$$f[w] \equiv \sum_{t=0}^{N-1} \langle \mathbf{e}_w, \mathbf{e}_t \rangle f[t] = \tfrac{1}{\sqrt{N}} \sum_{t=0}^{N-1} f[t] e^{-i\frac{2\pi w}{N}t}. \qquad \text{(DFTA)}$$

This is the usual change-of-basis formula from the standard basis in the time domain $\{\mathbf{e}_t\}$ to the Discrete Fourier basis $\{\mathbf{e}_w\}$.

To convert a signal from the frequency domain back to the time domain, we use the Discrete Fourier transform synthesis equation:

$$f[t] \equiv \sum_{w=0}^{N-1} \langle \mathbf{e}_t, \mathbf{e}_w \rangle f[w] = \tfrac{1}{\sqrt{N}} \sum_{w=0}^{N-1} f[w] e^{i\frac{2\pi w}{N}t}. \qquad \text{(DFTS)}$$

Again, this inverse change-of-basis transformation can be understood a special case of the standard change-of-basis formula that we learned in Section 5.3 (see page 198).

Sampling signals The *sampling rate* is an important practical consideration to take into account when converting analog signals to digital form. Usually the samples of the continuous signal $f(t)$ are taken at uniform time intervals spaced by time Δt. The *sampling rate* or *sampling frequency* f_s is defined as the inverse of the time between samples: $f_s = \frac{1}{\Delta t}$. For good performance, we must choose the sampling frequency f_s to be at least double that of the higher frequency of interest in the signal we're sampling.

For example, the upper limit of the human hearing range is 20 kHz, therefore CD digital audio recordings are sampled at $f_s = 44.1$ kHz. This means one second of audio recoding corresponds to 44 100 discrete audio samples, which we'll denote as a sequence $f[0]$, $f[1]$, $f[2]$, ..., $f[44099]$. The sample $f[0]$ corresponds to the value of the signal at $t = 0$, $f[1]$ corresponds to the the value of $f(t)$ at the next sampling time $t = \frac{1}{f_s}$, and so on for the rest of the samples $f[t] \equiv f(t\frac{1}{f_s})$.

Digital signal processing

Fourier transforms play a central role in digital signal processing. Many image compression and sound compression algorithms depend on the Discrete Fourier transform. For example `mp3` compression works by cutting up a song into short time segments, transforming these segments to the frequency domain using the Discrete Fourier transform, and then "forgetting" the high frequencies components.

Below we give a high-level overview of the pipeline of transformations used for `mp3` encoding and playback. We'll assume we're starting from a digital song recoding in the `wav` format. A `wav` file describes the song's sound intensity as a function of time, or we can say the song is represented in the time basis $[\texttt{song}]_t$.

The idea behind `mp3` encoding is to cut out the frequency components of a song that we cannot hear. The field of *psychoacoustics* develops models of human hearing which can inform us of which frequencies can be dropped without degrading the song's playback much. Without going into details, we'll describe the human psychoacoustic processing as a projection in the frequency domain ${}_f[\Pi]_f$. The overall `mp3` encoding and playback pipeline can be described as follows:

$$\underbrace{[\texttt{song}']_t = {}_t[\mathbb{1}]_f}_{\texttt{mp3 playback}} \underbrace{{}_f[\Pi]_f \; {}_f[\mathbb{1}]_t[\texttt{song}]_t}_{\texttt{mp3 encoding}}.$$

The sound information flows from right to left in the above equation. First we use the Fourier transform, denoted ${}_f[\mathbb{1}]_t$, to transforms the song from the time-basis to the frequency-basis, where the psychoacoustic projection is applied ${}_f[\Pi]_f$. An `mp3` file is the frequency-domain representation of the song, after applying the projection: $[\texttt{song}']_f = {}_f[\Pi]_f \; {}_f[\mathbb{1}]_t[\texttt{song}]_t$. Compression ratios up to a factor of 10 can be achieved using this approach: a 50 MB `wav` file can be compressed to a 5 MB `mp3` file. During playback, the inverse change-of-basis ${}_t[\mathbb{1}]_f$ is used to transforms the song from the frequency-basis back to the time-basis.

Obviously some level of sound degradation occurs in this process, and the song you'll hear during playback $[\texttt{song}']_t$ will be different from the original $[\texttt{song}]_t$. If good models of human hearing are used, the differences should be mostly imperceptible. Crucially to the `mp3` encoding process, the psychoacoustic models are applied in the frequency domain. This is an example where the Fourier transform as a building block for a more complex procedure.

Discussion

In this section we learned about three different Fourier transformations: the Fourier series, the Fourier transform, and the Discrete Fourier transform. All the scary-looking formulas which we saw can understood as special cases of the same idea: Fourier transformations are different types of change-of-basis to a set of orthogonal functions of the form $e^{i\omega t}$, which form a basis in the frequency domain.

It may seem like the Fourier series with its sine and cosine terms is a different kind of beast, but only until you recall Euler's formula $e^{i\theta} = \cos\theta + i\sin\theta$ (see page 106). Using Euler's formula we can express

the Fourier series of a function as a sequence of complex coefficients $c_n \in \mathbb{C}$ that contain the combined information of both cos- and sin-like components of a periodic signal $f(t)$. The complex Fourier coefficients c_n are obtained using the formula

$$c_n = \int_0^T \left\langle e^{i\frac{2\pi n}{T}t}, \mathbf{e}_t \right\rangle f(t)\,dt = \frac{1}{T}\int_0^T e^{-i\frac{2\pi n}{T}t} f(t)\,dt.$$

Problem **P8.3** will ask you to verify the connection between the Fourier series of complex exponentials and the Fourier series of sines and cosines discussed above.

The Fourier change of basis is a general principle that can be applied in many different contexts. The transforms discussed above used the family of orthogonal functions $e^{i\omega t}$, but there are other sets of orthogonal functions that can be used as alternate bases for different applications. To mention but a few, we have the Bessel functions J_λ, the spherical harmonics $Y_{mn}(\theta, \psi)$, the Legendre polynomials $P_n(x)$, the Hermite polynomials $H_n(x)$, and the Laguerre polynomials $L_n(x)$. All these families of functions form orthogonal sets on different intervals and with respect to different inner products.

Links

[Visualizations of Fourier synthesis of a square wave signal]
`http://codepen.io/anon/pen/jPGJMK/`
`http://bgrawi.com/Fourier-Visualizations/`

[Excellent video tutorial about digital audio processing]
`http://xiph.org/video/vid2.shtml`

[Website with animations that explain signal processing concepts]
`http://jackschaedler.github.io/circles-sines-signals/`

[The Wikipedia pages of the three Fourier transforms described]
`https://en.wikipedia.org/wiki/Fourier_series`
`https://en.wikipedia.org/wiki/Fourier_transform`
`https://en.wikipedia.org/wiki/Discrete_Fourier_transform`

[A nice discussion on math.stackexchange.com]
`https://math.stackexchange.com/questions/1002/`

[Orthogonal polynomials and generalized Fourier series]
`http://math24.net/orthogonal-polynomials.html`

Exercises

E8.11 Recall the functions $\mathbf{e}_n(x) \equiv \sin(\frac{n\pi}{L}x)$ that can be used to describe all vibrations of a guitar string of length L. Verify that $\mathbf{e}_1(x)$

and $\mathbf{e}_2(x)$ are orthogonal functions by computing the inner product $\langle \mathbf{e}_1(x), \mathbf{e}_2(x) \rangle$. Use the inner product definition from page 378.

Hint: You might find the double angle formula from page 64 useful.

E8.12 Explain why the Fourier series $(a_0, a_1, b_1, a_2, b_2, \ldots)$ of a periodic function $f(t)$ contains a coefficient a_0 but not a coefficient b_0.

Discussion

We have only scratched the surface of all the possible problem domains that can be modelled using linear algebra. The topics covered in this chapter are a small sample of the range of scientific, computing, and business activities that benefit from the use of matrix methods and understanding of vector spaces. Linear algebra allows you to perform complex numerical procedures using a high level of abstraction.

Normally, a linear algebra class should end here. We've covered all the information you need to know about vectors, matrices, linear transformations, vector spaces, and also discussed several applications. Feel free to close this book now, feeling content that your valiant effort to learn linear algebra is done. But perhaps you would like to learn some further topics and make use of your linear algebra skills? If this sounds interesting, I encourage you to keep on reading, as there are two more chapters of cool "optional material" ahead.

Chapter 9 covers the basics of probability theory, Markov chains, and the idea behind Google's `PageRank` algorithm for classifying web pages. Probability theory is not directly related to linear algebra, but the applications we'll discuss make heavy use of linear algebra concepts.

The laws of quantum mechanics govern physics phenomena at the femto-scale—think individual atoms. It's a common misconception to assume quantum laws are somehow mysterious or counterintuitive, and perhaps they are for people without the appropriate math background. Chapter 10 contains a concise introduction to the principles of quantum mechanics specifically tailored to people who know linear algebra. The material is adapted from lectures the author prepared for a graduate-level introductory course, so no dumbing down or simplification will be done. With your background in linear algebra, you can handle the real stuff.

Links

You'll find the following resources useful if you want to learn more about linear algebra applications.

[Three compilations of linear algebra applications]
`http://aix1.uottawa.ca/~jkhoury/app.htm`
`https://medium.com/@jeremyjkun/633383d4153f`
`http://isites.harvard.edu/fs/docs/icb.topic1011412.files/applications.pdf`

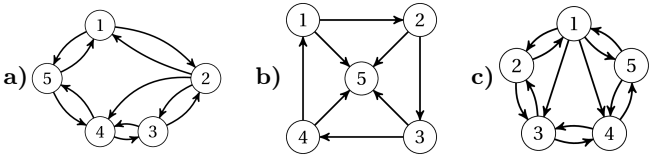
[A document that describes many applications in detail]
`http://gwu.geverstine.com/linearalgebra.pdf`

[Thirty-three miniatures: algorithmic applications of linear algebra]
`http://kam.mff.cuni.cz/~matousek/stml-53-matousek-1.pdf`

[A book about linear algebra applications to data science]
`http://amazon.com/Data-Science-from-Scratch/dp/149190142X`

8.12 Applications problems

P8.1 Find the adjacency matrix representation of the following graphs:



P8.2 For each of the graphs in **P8.1**, find the number of ways to get from vertex 1 to vertex 3 in two steps or less.

Hint: Obtain the answer by inspection or by looking at the appropriate entry of the matrix $\mathbb{1} + A + A^2$.

P8.3 Consider the signal $f(t)$ that is periodic with period T. The coefficients of the *Complex Fourier series* are defined using the formula

$$c_n = \frac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt.$$

Show that $c_n = a_n - ib_n$ where a_n and b_n are the coefficients of the regular Fourier series for $f(t)$, defined in terms of cosines and sines.

Hint: Obtain the real and imaginary parts of c_n using Euler's formula.

Chapter 9

Probability theory

In this chapter we'll use linear algebra concepts to explore the world of probability theory. Think of this as a "bonus" chapter because the topics we'll discuss are not normally part of a linear algebra course. Given the general usefulness of probabilistic reasoning and the fact you have all the prerequisites, it would be a shame *not* to learn a bit about probability theory and its applications, hence this chapter.

The structure of the chapter is as follows. In Section 9.1 we'll discuss probability distributions, which are mathematical models for describing random events. In Section 9.2 we'll introduce the concept of a *Markov chain*, which can be used to characterize the random transitions between different states of a system. Of the myriad of topics in probability theory, we've chosen to discuss probability distributions and Markov chains because they correspond one-to-one with vectors and matrices. This means you should feel right at home. In Section 9.3 we'll show how to use matrices to represent links between nodes in any network, and describe Google's `PageRank` algorithm for ranking webpages.

9.1 Probability distributions

Many phenomena in the world around us are inherently unpredictable. When you throw a six-sided die, one of the outcomes $\{1, 2, 3, 4, 5, 6\}$ will result, but you don't know which one. Similarly when you toss a coin, you know the outcome will be either `heads` or `tails` but you can't predict which outcome will result. Probabilities can be used to model real-world systems. For example, we can build a probabilistic model of hard drive failures using past observations. We can then calculate the probability that your family photo albums will survive the next five or ten years. Backups my friends, backups.

Probabilistic models can help us better understand random events. The fundamental concept in probability theory is the concept of a *probability distribution*, which describes the likelihood of the different outcomes of a random event. For example, the probability distribution for the roll of a fair die is $p_X = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^\mathsf{T}$, and the probability distribution for a coin toss is $p_Y = (\frac{1}{2}, \frac{1}{2})^\mathsf{T}$. Each entry of a probability distribution corresponds to the *probability mass* of a given outcome. A probability mass is a nonnegative number between 0 and 1. Furthermore, the sum of the entries in a probability distribution is 1. These two conditions are known as the *Kolmogorov axioms* of probability.

Strictly speaking, understanding linear algebra is not required for understanding probability theory. However, vector notation is very useful for describing probability distributions. Using your existing knowledge of vectors and the rules for matrix multiplication will allow you to understand many concepts in probability theory in a very short time. Probabilistic reasoning is a very useful so this digression is totally worth it.

Random variables

A random variable X is associated with a probability distribution p_X. Before we formally define the notion of a probability distribution, we must introduce some formalism. We denote by $\mathcal{X}$ (calligraphic X) the set of possible outcomes of the random variable X. A *discrete* random variable has a finite set of possible outcomes. For example, we can describe the outcome of rolling a six-sided die as the random variable $X \in \mathcal{X}$. The set of possible outcomes is $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$. The number of possible outcomes is six: $|\mathcal{X}| = 6$.

We can describe the randomness that results in a coin toss as a random variable $Y \in \{\texttt{heads}, \texttt{tails}\}$. The possible outcomes of the coin toss are $\mathcal{Y} = \{\texttt{heads}, \texttt{tails}\}$. The number of possible outcomes is two: $|\mathcal{Y}| = 2$.

In the case of the hard disk failure model, we can define the random variable $L \in \mathbb{N}$ that describes the years of lifetime of a hard disk before it fails. Using the random variable L we can describe interesting scenarios and reason about them probabilistically. For example, the condition that a hard disk will function correctly at least 8 years can be described as $L \geq 8$.

The set of all possible outcomes of a random variable is called the *sample space*. The probability distribution of a random variable specifies the *probability mass* to each of the possible outcomes in the sample space.

Probability distributions

The probability distribution p_X of a discrete random variable $X \in \mathcal{X}$ is a vector of $|\mathcal{X}|$ nonnegative numbers whose sum equals one. Using mathematically precise notation, we write this definition as

$$p_X \in \mathbb{R}^{|\mathcal{X}|} \quad \text{such that} \quad p_X(x) \geq 0, \forall x \in \mathcal{X} \quad \text{and} \quad \sum_{x \in \mathcal{X}} p_X(x) = 1.$$

A probability distribution is an ordinary vector in $\mathbb{R}^{|\mathcal{X}|}$ that satisfies two special requirements: its entries must be nonnegative and the sum of the entries must be one. Starting from these simple principles for describing probabilistic phenomena, we can build a rich science of random events, expectations, statistics, and develop methods for predicting the likelihood of future events. In the remainder of this section, we'll learn about probabilistic reasoning concepts through a series of examples.

Example 1 Recall the random variables X that describes the outcome of rolling a six-sided die. Assuming the die is fair, the probability distribution of the random variable X is

$$p_X = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix} \begin{matrix} \leftarrow p_X(1) \\ \leftarrow p_X(2) \\ \leftarrow p_X(3) \\ \leftarrow p_X(4) \\ \leftarrow p_X(5) \\ \leftarrow p_X(6) \end{matrix}.$$

Note the unusual notation for referring to the entries of the probability distribution p_X:

$$\text{the } x^{\text{th}} \text{ element of } p_X \quad \equiv \quad p_X(x) \quad \equiv \quad \Pr(\{X = x\}).$$

Using the normal vector notation, we could denote $p_X(x)$ as p_{Xx}, but using two subscripts might be confusing. When the random variable X is clear from the context, we can use the lighter notation p_x to denote the entry $p_X(x)$.

The notation $\Pr(\{X = x\})$ is the most precise. The number $p_X(x)$ corresponds to the probability denoted Pr of the *event* that a random draw from X results in the outcome x, denoted $\{X = x\}$. We postpone the discussion about random events until the next section, and continue with more examples of random variables and associated probability distributions.

Example 2 The random variable Y which describes the outcome of a coin toss is

$$p_Y = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \begin{matrix} \leftarrow p_Y(\texttt{heads}) \\ \leftarrow p_Y(\texttt{tails}) \end{matrix}$$

Strictly speaking the probability distribution p_Y is a function of the form $p_Y : \{\texttt{heads}, \texttt{tails}\} \to [0, 1]$, but for the sake of notational convenience, we associate the probability of `heads` and `tails` with the first and second entries of a two-dimensional vector p_Y. This is a book about linear algebra, so everything must be turned into a vector!

Interpretations of probability theory

One approach for understanding probability theory is to think about probabilities as describing relative frequencies of occurrence for the different possible outcomes. The quantity $p_X(a)$ represents the proportion of outcomes of the event $\{X = a\}$ among all possible outcomes of X.

This suggest an approach for estimating the probability mass of each of the possible outcomes. To characterize some real-world phenomenon as the random variable X, we estimate the probability distribution of X by repeating the real-world phenomenon n times, where n grows to infinity. The probability mass of each of outcome $a \in \mathcal{X}$ is defined as the following limit:

$$p_X(a) \equiv \frac{N(\{X = a\}, n)}{n}, \quad n \to \infty,$$

where $N(\{X = a\}, n)$ is number of times the outcome a occurs during n repetitions. If I tell you the probability mass of outcome $\{X = a\}$ is 0.3, this tells you that if you take 1000 draws from the random variable X, you can expect approximately 300 of those draws will result in outcome $\{X = a\}$.

This is called the *frequentist* point of view about probabilities, which is well suited for thinking about events that can be repeated many times. In practice we never really have the leisure to repeat events an infinite number of times to obtain the exact probabilities, so you have to think of this frequentist definition as a thought experiment—not something you would do in practice.

Another way to think about probabilities is as the state of *knowledge* or *belief* about the world. Instead of describing some objective reality, the random variable X its probability distribution p_X represents our state of knowledge about the real-world phenomenon that X describes. Since p_X represents our state of knowledge about the random variable X, it makes sense to update the distribution p_X as we learn new facts. This is called the *Bayesian* point of view, named

after Thomas Bayes who was an 18^{th} century statistician. Consider the following example of Bayesian-style reasoning. You're given a coin and asked to come up with a probability distribution that describes its chances of falling `heads` or `tails`. Initially you have no information about the coin being biased either way, so it would make sense to start win the initial belief that the coin is fair. This is called the *prior belief* or simply *prior*. If you toss the coin a few times and obtain much more `heads` than `tails`, you can then update your belief about the coin's probability distribution to take into account the observed data. The specific technique for updating the prior belief in the light of new observations is called *Bayes' rule*.

Both the frequentist and Bayesian points of view lead to useful techniques for modelling random events. Frequentist methods are concerned with drawing principled conclusions given a set of empirical observations. Bayesian models are generally more flexible and allow us to combine empirical observations with priors that encode the domain knowledge provided by experts. The frequentist approach is useful when you want to analyze data, prepare reports, and come up with hard numbers about the data. The Bayesian approach is more useful for building machine learning applications like voice recognition.

Conditional probability distributions

Probability theory allows us to model dependencies between random variables. We use *conditional probability distributions* to describe situations where one random variable depends on another. Consider the random variable X whose random outcomes depend on another variable Y. To describe this probability dependence we must specify the probability distributions of X for each of the possible values of Y. This information is expressed as a conditional probability distribution:

$$p_{X|Y}(x|y) \equiv \Pr(\{X = x\} \text{ given } \{Y = y\}).$$

The vertical bar is pronounced "given," and it is used to separate the unknown random variable from the random variable whose value is known. The distribution $p_{X|Y}$ satisfies the conditions for a probability distribution for all $y \in \mathcal{Y}$:

$$p_{X|Y} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{Y}|} \text{ s.t. } p_{X|Y}(x|y) \geq 0 \text{ and } \sum_{x\in\mathcal{X}} p_{X|Y}(x|y) = 1, \forall y \in \mathcal{Y}.$$

It is natural to represent $p_{X|Y}$ as a $|\mathcal{X}| \times |\mathcal{Y}|$ matrix, with each column representing the probability of X for a given value of Y.

Example 3 Consider a probability distribution with six possible outcomes obtained by rolling one of two different dice. If we're given

the fair die, the probability distribution is $p_{X_f} = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^\mathsf{T}$. If we're given the biased die the probability distribution is $p_{X_b} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0)^\mathsf{T}$. Introducing the conditioning variable Y that describes which die is used, we can express the situation with the two dice as the following conditional probability distribution:

$$p_{X|Y} \equiv \begin{bmatrix} \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & 0 \\ \frac{1}{6} & 0 \end{bmatrix}.$$

The first first column corresponds to the fair die $Y = f$, and the second column corresponds to the biased die $Y = b$.

Combining the probability distributions for the two dice p_{X_f} and p_{X_b} into a single conditional distribution $p_{X|Y}$ allows us to construct more complex probabilistic structures, as illustrated in the next example.

Example 4 Consider an experiment in which the outcome of a coin toss Y decides which die we'll throw—the fair die or the biased die. Suppose the coin is biased with $p_Y = (\frac{3}{4}, \frac{1}{4})^\mathsf{T}$. If the outcome of the coin toss is `heads`, we roll the fair die. If the coin toss gives `tails`, we roll the biased die. We're interested in describing the random variable X which corresponds to a die roll in which the fair die is used $\frac{3}{4}$ of the time and the biased die is used $\frac{1}{4}$ of the time. What is the probability distribution p_X for X?

To model this situation, we combine the "which die" probability distribution $p_Y = (\frac{3}{4}, \frac{1}{4})^\mathsf{T}$ and the conditional probability distribution $p_{X|Y}$ obtained in previous example:

$$\begin{aligned} p_X(x) &\equiv \sum_{y \in \mathcal{Y}} p_{X|Y}(x|y) p_Y(y) \\ &= p_{X|Y}(x|f) p_Y(\texttt{heads}) + p_{X|Y}(x|b) p_Y(\texttt{tails}) \\ &= p_{X|Y}(x|f) \tfrac{3}{4} \qquad\qquad + p_{X|Y}(x|b) \tfrac{1}{4} \\ &= \tfrac{3}{4}(\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6})^\mathsf{T} + \tfrac{1}{4}(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, 0, 0)^\mathsf{T} \\ &= (\tfrac{3}{16}, \tfrac{3}{16}, \tfrac{3}{16}, \tfrac{3}{16}, \tfrac{1}{8}, \tfrac{1}{8})^\mathsf{T}. \end{aligned}$$

The probabilistic mixture of two random events corresponds to a linear combination. When you head "linear combination" you immediately think "matrix-vector product representation," right? Indeed, we

can also express p_X as a matrix-vector product between the conditional probability distribution $p_{X|Y}$ (a matrix) and distribution p_Y (a vector):

$$p_X = \begin{bmatrix} \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{4} \\ \frac{1}{6} & 0 \\ \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} \frac{3}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{3}{16} \\ \frac{3}{16} \\ \frac{3}{16} \\ \frac{3}{16} \\ \frac{1}{8} \\ \frac{1}{8} \end{bmatrix}.$$

You can verify that $\sum_x p_X(x) = 1$ as expected.

Note how easy it is to compose conditional probability distributions to describe complicated random events in terms of the matrix-vector product. The notion of a conditional probability distribution is a fundamental building block of probability theory. Many modern machine learning techniques use probabilistic models constructed from conditional probability distributions.

Discussion

We described the basic notions in probability theory like random variables, and discussed probability distributions and conditional probability distributions. These concepts are the bread-and-butter concepts of probabilistic reasoning. Let's take a minute to recap and summarize the new material and link it back to vectors and linear transformations, which are the main subjects of the book. The probability distribution of a discrete random variable is a vector of real numbers. The probability distribution describing the outcome of a six sided die is a six-dimensional vector $p_X \in \mathbb{R}^6$. Probability distributions are vectors that obey some extra constraints. Each entry must be a positive number and the sum of the entries in the vector must be one.

Conditional probability distributions are mappings that describe how a set of "given" variables influence the probabilities of a set of "outcome" random variables. Conditional probability distributions can be represented as matrices, where each column of the matrix contains the outcome distribution for one of the values of the given random variable. A conditional probability distribution with five possible outcomes, conditioned on a "given" variable with ten possible states is represented as a 5×10 matrix, whose columns all sum to one.

Conditional probability distributions are a very powerful tool for modelling real-world scenarios. For example, you could describe a noisy communication channel as a conditional probability distribution of the channel outputs given each of the possible channel inputs. Many

machine learning algorithms involve the characterization, estimation, and exploitation of conditional probability distributions—collectively called the *model*.

In the next section we'll learn about an application of conditional probability distribution for describing the state of a system undergoing random transitions between a number of possible states. This is an interesting special case of a conditional probability distribution for which conditioning and outcome random variable are of the same dimension. Indeed, the outcome variable and the conditional space both represent states of the same system at different times.

Links

[Discussion on the Bayesian way of thinking about probabilities]
`https://en.wikipedia.org/wiki/Bayesian_probability`

[Detailed discussion about Bayes rule from first principles]
`http://yudkowsky.net/rational/bayes`
`http://yudkowsky.net/rational/technical`

Exercises

E9.1 Do the following vectors represent probability distributions:

$$\textbf{a)}\ \left(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}\right)^\mathsf{T} \qquad \textbf{b)}\ \left(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\right)^\mathsf{T} \qquad \textbf{c)}\ (0.3, 0.3, -0.1, 0.5)^\mathsf{T}$$

9.2 Markov chains

So far we talked about random events without any reference to the flow of time. In this section we'll combine the idea of random variables with the notion of time. A *random process* is a model of a system that undergoes transitions between states over time. The state of the system at time t is described by a random variable X_t. We assume the state of the system depends on the previous states through a conditional probability distribution:

$$p_{X_{t+1}|X_t X_{t-1} \ldots X_0}(x_{t+1}|x_t x_{t-1} \ldots x_0).$$

This means the random variable X_{t+1} depends on the previous states of the system: X_t, X_{t-1}, X_{t-2}, ..., X_0. Studying such history-dependent processes is a formidable task because of the myriad of possible influences from past states. Faced with this complexity, we can make a useful simplifying assumption and consider *memoryless* random processes.

A *Markov process* or *Markov chain* is a random process for which the transition probabilities for the next state depend only on the current state and not on states before it:

$$p_{X_{t+1}|X_tX_{t-1}\ldots X_0}(x_{t+1}|x_tx_{t-1}\ldots x_0) = p_{X_{t+1}|X_t}(x_{t+1}|x_t).$$

The next state X_{t+1} depends only on the current X_t and not on the pas history of the process: X_{t-1}, X_{t-2}, ..., X_0. Thus, a Markov chain can be fully described by a conditional probability distribution $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$, which describes the probability of the next state of the system given the current state.

Markov chains are an extremely versatile model for many real-world systems. We don't have the space here to cover the topic in full detail, but I plan to introduce the basic notions so you'll know about them. First we'll show the connection between the evolution of a Markov chain and the matrix product. Next we'll see how your eigenvalue-finding skills can be used to compute an important property of Markov chains. Understanding Markov chains is also necessary background material for understanding Google's `PageRank` algorithm, which we'll discuss in Section 9.3.

Example

Consider three friends who are kicking around a football in the park: Alice, Bob, and Charlie. When Alice gets the ball, the makes a pass to Bob 40% of the time, a pass to Charlie 40% of the time, or holds on to the ball 20% of the time. Bob is a bit of a greedy dude: when he gets the ball he holds on to it 80% of the time, and is equally likely to pass to Alice or Charlie in to remaining 20% of the time. When Charlie gets the ball he's equally likely to pass the ball to Alice, Bob, or keep it for himself. Assume these friends kick the ball around for a very long time (hundreds of passes) and you observe them at some point. What is the probability that each players will be in possession of the ball at the instant when you observe them?

We can model the ball possession as a Markov process with three possible states: $\mathcal{X} = \{A, B, C\}$, each describing the "state" of the football game when Alice, Bob, or Charlie has the ball. The transition probabilities $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$ describe how the next ball possession x_{t+1} depends on the previous state of the ball possession x_t. The transition matrix of the Markov chain in our current example is

$$p_{X_{t+1}|X_t} \equiv \begin{bmatrix} 0.2 & 0.1 & 0.\overline{3} \\ 0.4 & 0.8 & 0.\overline{3} \\ 0.4 & 0.1 & 0.\overline{3} \end{bmatrix} \equiv M.$$

For consistency with the notation of conditional probability distributions, we refer to the coefficients of this matrix M as $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$. The "given" variable x_t selects the column of the matrix M and the different entries in this column represent the transition probabilities for that state. Using the matrix M and some basic linear algebra techniques we can calculate the probability of finding the ball in any given player after many iterations of the "pass the ball" Markov process.

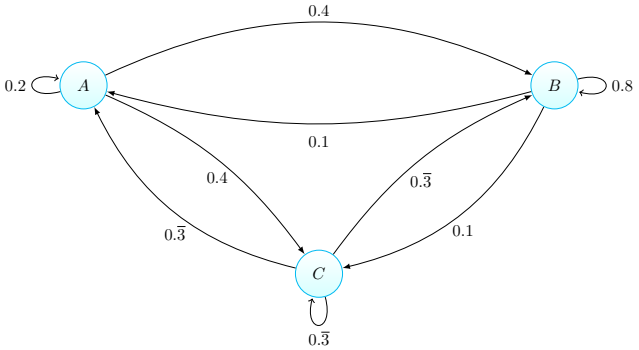


Figure 9.1: Graphical representation of the transition probabilities between the three states: "Alice has the ball," "Bob has the ball," and "Charlie has the ball."

Let's walk through an example calculation, in which we assume the ball starts with Alice initially. Since we know for certain that Alice has the ball at $t = 0$, the initial state of the system is described by the probability distribution $p_{X_0} = (1, 0, 0)^\mathsf{T}$, with 100% of the weight on Alice. The probability of finding the ball with each player after one time step is obtained by multiplying the initial probability vector p_{X_0} by the matrix M:

$$
\begin{aligned}
p_{X_1} &= M p_{X_0} \\
&= \begin{bmatrix} 0.2 & 0.1 & 0.\overline{3} \\ 0.4 & 0.8 & 0.\overline{3} \\ 0.4 & 0.1 & 0.\overline{3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix}.
\end{aligned}
$$

All the "where is the ball" probability mass started out on Alice, but after one time step it spread to Bob and Charlie, according to Alice's expected passing behaviour (the first column in the transition matrix).

We can continue this process to obtain the probability of ball possession after two time steps. We simply multiply the probability vector p_{X_1} by the transition matrix M, which is the same as multiplying

the initial state p_{X_0} by M twice:

$$\begin{aligned} p_{X_2} &= MMp_{X_0} = Mp_{X_1} \\ &= \begin{bmatrix} 0.2 & 0.1 & 0.\overline{3} \\ 0.4 & 0.8 & 0.\overline{3} \\ 0.4 & 0.1 & 0.\overline{3} \end{bmatrix} \begin{bmatrix} 0.2 & 0.1 & 0.\overline{3} \\ 0.4 & 0.8 & 0.\overline{3} \\ 0.4 & 0.1 & 0.\overline{3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.21\overline{3} \\ 0.5\overline{3} \\ 0.25\overline{3} \end{bmatrix}. \end{aligned}$$

The probability that Alice is holding the ball after two time steps is $0.21\overline{3}$. This number represents the combination of three different "paths" of starting from the state $x_0 = A$ and coming to the state $x_2 = A$ at $t = 2$. Either Alice kept the ball for two time steps ($x_1 = A$), or Alice passed the ball to Bob ($x_1 = B$) and then he passed it back to her, or Alice passed the ball to Charlie ($x_1 = C$) and Charlie passed it back her. By computing the sum of the products of the probabilities of the events on each path, we arrive at the value

$$\begin{aligned} p_{X_2|X_0}(A|A) &= p_{X_2|X_1}(A|A)\ p_{X_1|X_0}(A|A) \\ &\quad + p_{X_2|X_1}(A|B)\ p_{X_1|X_0}(B|A) \\ &\quad + p_{X_2|X_1}(A|C)\ p_{X_1|X_0}(C|A) \\ &= 0.2 \times 0.2 + 0.1 \times 0.4 + 0.\overline{3} \times 0.4 \\ &= 0.21\overline{3}. \end{aligned}$$

Observe the comparative simplicity of the multiplication by the transition matrix, which "takes care" of these three different paths automatically. Do you have to manually think about all possible paths? Nope. Just multiply the state by the transition matrix and you obtain the state probabilities in the next time step.

Thanks to the power of the Markov chain construction, we can carry the "ball possession" probabilities vector forward through time as far as we want to. To calculate the random state $p_{X_{t+1}}$, multiply by the previous state vector p_{X_t} by the Markov transition probabilities matrix M. The probability distribution of ball possession after three time steps is

$$p_{X_3} = Mp_{X_2} = MMp_{X_1} = MMMp_{X_0} = M^3 p_{X_0} = \begin{bmatrix} 0.180\overline{4} \\ 0.596\overline{4} \\ 0.223\overline{1} \end{bmatrix}.$$

Continuing this process, we can obtain the probability state vector after 10 and after 20 times steps of the Markov chain:

$$p_{X_{10}} = M^{10} p_{X_0} = \begin{bmatrix} 0.161\ldots \\ 0.645\ldots \\ 0.193\ldots \end{bmatrix} \quad \text{and} \quad p_{X_{20}} = M^{20} p_{X_0} = \begin{bmatrix} 0.1612903\ldots \\ 0.6451612\ldots \\ 0.193548\ldots \end{bmatrix}.$$

Observe the probability distribution seems to approach a steady state. This is the "long term" probability question we're looking to answer in this example.

Stationary distribution

If the evolution of a Markov chain continues for long enough, the probability vector will converge to a stable distribution p_{X_∞} that remains unchanged when multiplied by M:

$$Mp_{X_\infty} = p_{X_\infty}.$$

This is called the *stationary distribution* of the Markov chain. Observe that p_{X_∞} is an eigenvector of the matrix M with eigenvalue $\lambda = 1$.

The convergence to a unique stationary distribution is a fundamental property of Markov chains. Assuming the Markov chain represented by M satisfies some technical conditions (that we won't go into), it will converge to a stationary distribution p_{X_∞}. Thus, if we want to find p_{X_∞} we just have to keep repeatedly multiplying by M until the distribution stabilizes:

$$p_{X_\infty} = \begin{bmatrix} 0.161290322580645\ldots \\ 0.645161290322581\ldots \\ 0.193548387096774\ldots \end{bmatrix} = M^\infty p_{X_0}.$$

The Markov chain will converge to the same stationary distribution p_{X_∞} regardless of the starting point p_{X_0}. The ball could have started with Bob $(0,1,0)^\mathsf{T}$ or with Charlie $(0,0,1)^\mathsf{T}$, and after running the Markov chain for long enough we would still arrive at the stationary distribution p_{X_∞}.

Since we know the stationary distribution is an eigenvector of M with eigenvalue $\lambda = 1$, we can use the usual eigenvector-finding techniques to obtain p_{X_∞} directly. We find the eigenvector by solving for $\vec{v}$ in $(M - \mathbb{1})\vec{v} = \vec{0}$. The answer is $p_{X_\infty} = (\frac{5}{31}, \frac{20}{31}, \frac{6}{31})^\mathsf{T}$. A nice benefit of this approach is that we obtain the exact analytic expression for the answer.

Discussion

Markov chains, despite being the simplest type of random process, have countless applications in physics, speech recognition, information processing, machine learning, and many other areas. Their simple memoryless structure and their intuitive representation as matrices make them easy to understand and easy to fit to many situations.

In the next section we'll describe a Markov chain model for people's Web browsing behaviour. The stationary distribution of this Markov chain serves to quantify the relative importance of webpages.

Links

[Awesome visual representation of states and transitions]
`http://setosa.io/blog/2014/07/26/markov-chains/index.html`

[More details about applications from wikipedia]
`https://en.wikipedia.org/wiki/Markov_chain`

Exercises

E9.2 After reading the section on Markov chains, you decide to research the subject further and get a book on Markov chains from the library. In the book's notation, Markov chains are represented using *right*-multiplication of the state vector: $\vec{v}' = \vec{v}B$, where $\vec{v}$ is the state of the system at time t, $\vec{v}'$ is the state at time $t+1$, and the matrix B represents the Markov chain transition probabilities.

Find the matrix B that corresponds to the transition probabilities discussed in Example 5. How is this matrix B related to the matrix M, which we used in Example 5?

$B = M^{\mathsf{T}}$.

E9.3 Use the `SymPy Matrix` method `.eigenvects()` to confirm the stationary probability distribution of the passing-the-ball Markov chain is indeed $\left(\frac{5}{31}, \frac{20}{31}, \frac{6}{31}\right)$.

Hint: To obtain the exact result, make sure you define the matrix A using fractional notation:

```
>>> M = Matrix([[ 2/10, 1/10, 1/3 ],
                [ 4/10, 8/10, 1/3 ],
                [ 4/10, 1/10, 1/3 ]])
```

Use `ev = M.eigenvects()[0][2][0]` to extract the eigenvector that correspond to the eigenvalue $\lambda = 1$, then normalize the vector by its 1-norm to make it a probability distribution `pinf = ev/ev.norm(1)`.

E9.4 Find the stationary probability distribution of the following Markov chain:

$$C = \begin{bmatrix} 0.8 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.6 \\ 0.1 & 0.5 & 0.2 \end{bmatrix}.$$

E9.5 Repeat the previous problem using the `nullspace` method for `SymPy` matrices to obtain p_{X_∞} by solving $(C - \mathbb{1})\vec{v} = \vec{0}$.

9.3 Google's PageRank algorithm

Consider the information contained in the **links between web pages**. Each link from Page A to Page B can be interpreted as a recommendation, by Page A's author, for the contents of Page B. In web-speak we say links from Page A to Page B are "sending eyeballs" to Page B, presumably because there is something interesting to see on Page B. These observations about "eyeball worthiness" are the inspiration behind Google's `PageRank` algorithm. We find a good summary of the idea behind `PageRank` in the 2001 patent application:

> A method assigns importance ranks to nodes in [...] the world wide web or any other hypermedia database. The rank assigned to a document is calculated from the ranks of documents citing it. In addition, the rank of a document is calculated from a constant representing the probability that a browser through the database will randomly jump to the document. — Patent `US6285999`

You may notice there is a weird-self referential thing going on in the definition. The rank of a document depends on the ranks of documents that link to it, but how do you discover the ranks of these documents? By calculating the ranks of documents that link to them, and so on and so forth. Don't worry if all this sounds confusing, it will all start to make sense very soon.

The random surfer model

Imagine someone who browses the Web by clicking on links randomly. We'll call this person Randy. Every time Randy opens his web browser he follows the following two strategies:

1. When visiting a webpage, he'll make a list of all the outbound links on that page and click on one of the links at random.
2. Randy randomly selects a page on the Web and goes to it.

Randy follows Strategy 1 90% of the time and Strategy 2 the remaining 10% of the time. In the unlikely event that he reaches a page with no outbound links, he'll switches to Strategy 2 and jumps to a random page.

You have to agree that Randy's behaviour is pretty random! This is a very simple model for the behaviour of users on the Web that assumes people either follow links blindly, or randomly jump to pages on the web. Simple as it be, this model allows us to capture an important aspect of "relative importance" of different webpages. In the next section we'll describe the random surfer model using the machinery of Markov chains.

The PageRank Markov chain

We want to construct a Markov chain that models the behaviour of Randy, the random Web surfer. The *state* in the Markov chain corresponds to the the webpage Randy is currently viewing, and the transition matrix will describes Randy's link following behaviour. This probabilistic reformulation will help us calculate the PageRank vector, which describes the "importance" of each webpage on the Web.

We'll now revisit the two random strategies used by Randy to browse the Web, and construct the appropriate Markov chain matrices to describe them. Note the dimension of the matrices is $n \times n$, where n is the number of webpages on the Web.

1. Suppose Randy is visiting webpage j, which has a total of N_j outbound links. Then the column for Page j should transition probability $\frac{1}{N_j}$ toward each of the pages that Page j links to. In the special case that Page j has zero outbound links, we'll fill column j with the uniform distribution $(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})^\mathsf{T}$. Thus, we can describe ij^{th} entry of the transition matrix for Strategy 1 as follows:

$$(M_1)_{ij} = \begin{cases} \frac{1}{N_j} & \text{if there's a link from Page } j \text{ to Page } i \\ \frac{1}{n} & \text{if Page } j \text{ has not outbound links} \\ 0 & \text{otherwise} \end{cases}$$

2. The transition matrix for Strategy 2 is much simpler. No matter which page Randy was visiting previously, we want him to jump to a random page on the Web, thus each column of the transition matrix M_2 will contain the uniform distribution over all pages $(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})^\mathsf{T}$. We can express this succinctly as:

$$M_2 = \frac{1}{n}\,\mathbb{J},$$

where $\mathbb{J}$ is a matrix that has ones in all its entries.

Recall that Randy uses Strategy 1 90% of the time and Strategy 2 10% of the time. The Markov chain M that describes his behaviour is a linear combination of the Markov chains for the two strategies:

$$M = (1-\alpha)M_1 + \alpha M_2 = (1-\alpha)M_1 + \frac{\alpha}{n}\,\mathbb{J},$$

where $\alpha = 0.1$ if we want to have the 90-10 mix of strategies, but in general α is a parameter we can tune for different applications.

The PageRank vector is the stationary distribution of the Markov chain M, which is defined through the equation $Mp_{X_\infty} = p_{X_\infty}$. We can

obtain p_{X_∞} by finding the eigenvector of M in the eigenspace $\lambda_1 = 1$, which is equivalent to solving the null space problem $(M - 1\mathbb{1})\vec{e} = \vec{0}$. We can also the stationary distribution p_{X_∞} by simply running the Markov chain for many iterations, until we see the probability distribution converge:

$$\{\, p_{X_\infty} \,\} = \mathcal{N}(M - \mathbb{1}) \qquad \Leftrightarrow \qquad p_{X_\infty} = M^\infty p_{X_0}.$$

Both approaches for finding p_{X_∞} are useful in different contexts. Solving the null space problem gives us the answer directly, whereas the second approach of iterating the Markov chain is more scalable for large graphs.

The `PageRank` value of a webpage is the probability that Randy will end up on a given page after running the Markov chain for a sufficiently long time. The `PageRank` vector p_{X_∞} captures an important aspect of well-connectedness or importance of webpages. Let's illustrate the entire procedure for calculating the `PageRank` vector in a simple graph.

Example: micro-Web

We'll now study the micro-Web illustrated in Figure 9.2. This is a simplified version of the link structure between webpages on the Web. Like a *vast* simplification! Instead of the billions of webpages of Web, the micro-Web has only eight webpages $\{1, 2, 3, 4, 5, 6, 7, 8\}$. Instead of the trillions of links between webpages, the micro-Web has only fourteen links $\{(1,2),\ (1,5),\ (2,3),\ (2,5),\ (3,1),\ (3,5),\ (4,5),\ (5,6),\ (5,7),\ (6,3),\ (6,7),\ (7,5),\ (7,6),\ (7,8)\}$. Still, this simple example is sufficient to illustrate the main idea of the `PageRank` algorithm. Scaling the solution from the case $n = 8$ to problem $n = 12\,000\,000$ is left as an exercise for the reader.

We'll first construct the transition matrix M_1 that corresponds to Strategy 1 (follow a random outbound link), then construct matrix M_2 for Strategy 2 (teleport to a random page), and construct a 90-10-weighted linear combination of M_1 and M_2, which is the `PageRank` Markov chain matrix.

The state of the Markov chain we want to constructs corresponds to the probability distribution of finding Randy on each of the eight pages. Let's say he is currently on Page 1, so the initial state of the Markov chain is $p_{X_0} = (1, 0, 0, 0, 0, 0, 0, 0)^\mathsf{T}$. Following Strategy 1, Randy is supposed to go to either Page 2 or Page 5. Since there are two possible outbound links, the probability mass of choosing one of them is $\frac{1}{2}$. Computing $M_1 p_{X_0}$ has the effect of "selecting" the first column of M_1, so we know what the first column of M_1 is. If we then "probe" M_1 with $p_{X_0} = (0, 1, 0, 0, 0, 0, 0, 0)^\mathsf{T}$, which corresponds

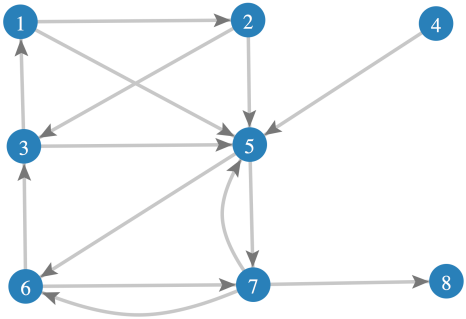


Figure 9.2: Graph showing the links between the pages on the micro-Web. Page 5 seems to be an important page because many pages link to it. Since Page 5 links to pages 6 and 7, these pages will probably get a lot of eyeballs too. Page 4 is the least important since there are no links to it. Page 8 is an example of the "unlikely" case of a webpage with no outbound links.

to Randy following links randomly starting at Page 2, we'll know what's in the second column of M_1. Continuing with the standard "probing" approach for building matrix representations, we find the rest of the entries in the transition matrix for Strategy 1:

$$M_1 = \begin{bmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{8} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 0 & 0 & \frac{1}{3} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{8} \end{bmatrix}.$$

You should convince yourself M_1 has the right entries by comparing the rest of the columns with link structure in Figure 9.2. Recall Strategy 1 handles the exceptional case of a page with no outbound links by jumping to a random page, since there are eight pages in total on the micro-Web, the uniform distribution over all pages is $(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})^\mathsf{T}$.

Do you recall the *adjacency matrix* representation for graphs we discussed in Section 8.4? You can obtain M_1 by taking the transpose of the adjacency matrix A then normalizing the columns so they become probability distributions.

The Markov chain for Strategy 2 is to jump to a random page on

the micro-Web:

$$M_2 = \frac{1}{8}\,\mathbb{J}_n = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}.$$

Note the new notation $\mathbb{J}_n$ which corresponds to a $n \times n$ matrix with ones in all its entries.

The `PageRank` Markov chain M is defined in terms of the matrices M_1 and M_2 and a mixture parameter, which we denote α. Choosing $\alpha = 0.1$ models a Randy who uses Strategy 1 90% of the time and Strategy 2 10% of the time:

$$\begin{aligned} M &= (1-\alpha)M_1 + \frac{\alpha}{8}\,\mathbb{J} \\ &= \frac{9}{10}\,M_1 + \frac{1}{80}\,\mathbb{J}, \end{aligned}$$

which is worth writing out in full detail:

$$M = \begin{bmatrix} \frac{1}{80} & \frac{1}{80} & \frac{37}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{8} \\ \frac{37}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{8} \\ \frac{1}{80} & \frac{37}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{37}{80} & \frac{1}{80} & \frac{1}{8} \\ \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{8} \\ \frac{37}{80} & \frac{37}{80} & \frac{37}{80} & \frac{73}{80} & \frac{1}{80} & \frac{1}{80} & \frac{5}{16} & \frac{1}{8} \\ \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{37}{80} & \frac{1}{80} & \frac{5}{16} & \frac{1}{8} \\ \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{37}{80} & \frac{37}{80} & \frac{1}{80} & \frac{1}{8} \\ \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{1}{80} & \frac{5}{16} & \frac{1}{8} \end{bmatrix}.$$

Notice the sprinkling of $\frac{1}{80}$s to fill the regions which had only zeros in M_1. This additional mixing is important for technical reasons. If we were to use just the Markov chain M_1, the probability distribution that describes Randy's state will not mix very well, but adding in a little bit of $\mathbb{J}_n$ guarantees the Markov chain will converge to its stationary distribution in the long term.

Having built the `PageRank` Markov chain for the micro-Web, we find its stationary distribution p_{X_∞}, which is the `PageRank` vector:

$$p_{X_\infty} = (0.08152, 0.05868, 0.1323, 0.02199, 0.2268, 0.1864, 0.2079, 0.08437)^\mathsf{T}$$

This vector was obtained by finding the vector in the $\lambda_1 = 1$ eigenspace of M. The author used `live.sympy.org` to obtain the result. See `bit.ly/microWebPR` for the commands required to solve the null space problem $(M - \mathbb{1})\,\vec{e} = \vec{0}$.

We can now use the entires of the `PageRank` vector to sort the pages of the micro-Web by their relative importance, as shown in Table 9.2.

Page ID	`PageRank`
Page 5	0.22678
Page 7	0.20793
Page 6	0.18642
Page 3	0.13229
Page 8	0.08437
Page 1	0.08152
Page 2	0.05868
Page 4	0.02199

Table 9.2: The `PageRank`s of the pages from the graph in Figure 9.2.

According their `PageRank` score, the top two page in the micro-Web are Page 5 with `PageRank` 0.22678 and Page 7 with `PageRank` 0.20793. Page 6 is not far behind with `PageRank` 0.18642. Looking back at Figure 9.2 we can confirm this ranking makes sense, since Page 5 has the most links pointing to it and it links to Page 6 and Page 7. The combined `PageRank` of these three pages is 0.62113, which means that over 62% of the time Randy's eyeballs will be viewing one of these three pages.

Discussion

The example we discussed above involved only eight webpages and an 8×8 Markov chain matrix. Imagine the size of the matrix we'll need to represent all the links between webpages on the Web. The web has upward of a billions webpages and contains trillions of hyperlinks. Imagine an $n \times n$ Markov chain matrix, where $n = 1\,000\,000\,000$. You'll need a lot of memory to store this matrix, and a lot of compute power to work with it. The computation of the `PageRank` vector for the entire Web can be performed using the *power iteration* method,

which finds the eigenvector for the largest eigenvalue of the matrix. Additionally, careful analysis of the data required to perform the algorithm can avoid the need to construct the impossibly-large $n \times n$ matrix. Thus the contribution of Larry Page and Sergey Brin go beyond the linear algebra insight, but also involve a clever approach for splitting the computation onto a cluster of computers. This is something to keep in mind when you apply your linear algebra knowledge to build stuff. You need to cultivate a mix of intuition, backed by mathematical tools, *and* engineering prowess to get things done. Go find the right linear combination of these resources and build you own $300B company.

Links

[The original `PageRank` paper]
`http://ilpubs.stanford.edu/422/1/1999-66.pdf`

[Further discussion about the `PageRank` algorithm]
`https://en.wikipedia.org/wiki/PageRank`

[The *power iteration* method for finding the `PageRank` eigenvector]
`https://en.wikipedia.org/wiki/Power_iteration`

Exercises

E9.6 Compute the `PageRank` of the network of webpages shown in Figure 8.3 (page 327).

Hint: Using `live.sympy.org`, you can define the stochastic matrix M using `Matrix`, then mix it with an all-ones matrix `ones(n,n)`. Use the `nullspace()` matrix method to obtain the eigenvector in the $\lambda = 1$ eigenspace.

9.4 Probability problems

P9.1 The probability of `heads` for a fair coin is $p = \frac{1}{2}$. The probability of getting `heads` n times in a row is given by the expression p^n. What is the probability of getting `heads` three times in a row?

P9.2 Consider the weather in a city which has "good" and "bad" years. Suppose the weather conditions over the years form a Markov chain where a good year is equally likely to be followed by a good or a bad year, while a bad year is three times as likely to be followed by a bad year as by a good year. Given that last year, call it Year 0, was a good weather year, find the probability distribution that describes the weather in Year 1, Year 2, and Year ∞.

Chapter 10

Quantum mechanics

Quantum mechanics was born out of need to explain certain observations in physics experiments, which could not be explained by previous physics theories. During the first half of the 20^{th} century, in experiment after experiment, quantum mechanics principles were used to correctly predict the outcomes of many atom-scale experiments. During the second half of the 20^{th} century, biologists, chemists, engineers, and physicists applied quantum principles to all areas of science. This process of "upgrading" classical models to quantum models leads to better understanding of reality and better predictions. Scientists like it that way.

In this chapter we'll introduce the fundamental principles of quantum mechanics. This little excursion into physics-land will expose you to the ideas developed by the likes of Bohr, Plank, Dirac, Heisenberg, and Pauli. You have all the prerequisites to learn about some fascinating 20^{th} century discoveries. All it takes to understand quantum mechanics is some knowledge of linear algebra (vectors, inner products, projections) and some probability theory (Chapter 9). Using the skill set you've built in this book, you can learn the main ideas of quantum mechanics at almost no additional mental cost.

This chapter is totally optional reading, reserved for readers who *insist* on learning about the quantum world. If you're not interested in quantum mechanics, it's okay to skip this chapter, but I recommend you check out Section 10.3 on *Dirac notation* for vectors and matrices. Learning Dirac notation serves as an excellent review of the core concepts of linear algebra.

10.1 Introduction

The fundamental principles of quantum mechanics can be explained in the space on the back of an envelope. These simple principles have far-reaching implications for many areas of science: physics, chemistry, biology, engineering, and many other fields of study. Scientists have adapted preexisting theories and models to incorporate quantum principles. Each field of study has its own view on quantum mechanics, and has developed a specialized language for describing quantum concepts. Before we introduce quantum mechanics in the abstract, let's look at some of the disciplines where quantum principles are used.

Physics Physicists use the laws of quantum mechanics as a toolbox to understand and predict the outcomes of atomic-scale physics experiments. By "upgrading" classical physics models to reflect the ideas of quantum mechanics, physicists (and chemists) obtain more accurate models that lead to better predictions.

For example, in a *classical* physics model, the motion of a particle is described by its position $x(t)$ and velocity $v(t)$ as functions of time:

- classical state: $(x(t), v(t))$, for all times t.

At any given time t, the particle is at position $x(t)$ and moving with velocity $v(t)$. Using Newton's laws of motion and calculus, we can predict the position and the velocity of a particle at all times.

In a quantum description of the motion of a particle in one-dimension, the state of a particle is represented by a *wave function* $|\psi(x,t)\rangle$, which is a complex-valued function of position x and time t:

- quantum state: $|\psi(x,t)\rangle$, for all times t.

At any given time t, the state of the particle corresponds to a complex-valued function of a real variable $|\psi(x)\rangle \in \{\mathbb{R} \to \mathbb{C}\}$. The wave function $|\psi(x)\rangle$ is also called the *probability-amplitude* function, since the probability of finding the particle at position x_a is proportional to the value of the squared-norm of the wave function:

$$\Pr(\{\text{particle position} = x_a\}) \quad \propto \quad \big||\psi(x_a)\rangle\big|^2.$$

Instead of having a definite position $x(t)$ as in the classical model, the position of the particle in a quantum model is described by a probability distribution calculated from its wave function $|\psi(x)\rangle$. Instead of having a definite momentum $p(t)$, the momentum of a quantum particle is another function calculated based on its wave function $|\psi(x)\rangle$.

Classical models provide accurate predictions for physics problems involving macroscopic objects, but fail to predict the physics of atomic-scale phenomena. Much of the 20^{th} century physics research effort has been dedicated to the study of quantum concepts like ground states, measurements, spin angular momentum, polarization, uncertainty, entanglement, and non-locality.

Computer science Computer scientists understand quantum mechanics using principles of information. Quantum principles impose a fundamental change to the "data types" used to represent information. Classical information is represented as *bits*, elements of the finite field of size two $\mathbb{Z}_2$:

- bit: $x = 0 \quad \text{or} \quad x = 1 \quad$ (two-level system)

In the quantum world, the fundamental unit of information is the *qubit*, which is a two-dimensional unit-length vector in a complex inner product space:

- qubit: $|x\rangle = \alpha|0\rangle + \beta|1\rangle \quad$ (unit-length vector in $\mathbb{C}^2$)

This change to the underlying information model requires reconsiderations of the fundamental information processing tasks like computation, data compression, encryption, and communication.

Philosophy Philosophers have also thought long and hard about the laws of quantum mechanics and what they imply about our understanding of the ultimate nature of reality. A question of central interest is how the *deterministic* laws of physics (clockwork-model of the universe) must be adapted to the quantum paradigm, in which experimental outcomes are inherently probabilistic. Another interesting question to consider is whether the quantum state $|\psi\rangle$ of physical system really exists, or if $|\psi\rangle$ is a representation of our knowledge about the system.

Many scientists are also interested in the philosophical aspects of quantum mechanics, but for the most part, professional scientists do not focus on interpretations. Scientists care only about which models lead to more accurate predictions. Since different philosophical interpretations of quantum phenomena cannot be tested experimentally, these questions are considered outside the scope of physics research.

Psychology There is even a psychology view on quantum mechanics. Some people try to apply quantum principles to the human mind. Neuroscience and quantum physics are two areas of science that are not fully understood, so supposing there is a link between the two is

not *provably* wrong, however current quantum psychology books offer nothing more than speculation. What is worse, it's speculation by people who know less about quantum mechanics than you will know by the end of this chapter.

Borrowing impressive-sounding words from physics can add an air of respectability to pop-psychology books, but don't be fooled by such tactics. We won't discuss this topic further in this book; I mentioned quantum psychology only to give you a heads-up about this nonsense. Next time you hear someone dropping the q-word in a context that has nothing to do with atomic-scale phenomena, call "bullshit" on them immediately and don't waste your time. They're just trying to bamboozle you.

Physical models of the world

Before we talk about quantum mechanics, it's important to define precisely the two conceptual "worlds" that we'll discuss:

- The **real world** is where physical experiments are performed.
- The **mathematical world** is a purely theoretical construct that aims to model certain aspects of the real world.

If a mathematical model is good, its predictions correspond closely to the behaviour of systems in the real world. Note there exist good models and better models, but no physics model can ever be "true" or "real" since, by it's very nature, it lives in the realm of formulas and equations and not of atoms and lasers. One physics model is considered more "true" than another only if it's better at predicting the outcomes of experiments. Coming up with physics models is a lot of fun. Anyone can be a physicist! These are the rules for constructing physical models: you're allowed to come up with any crazy mathematical model for describing nature, and if your model correctly predicts the outcomes of experiments, physicists will start using it.

We can make a further distinction among mathematical models, classifying them into two categories depending on the type of math they use:

- **Classical models** describe the world in terms of concepts like positions and velocities and the way system states evolve from one instant to the next is governed by deterministic laws.
- **Quantum models** describe systems in terms of vectors in complex vector spaces. The way systems evolve over time is governed by a deterministic laws, but we can only "observe" quantum systems by performing measurements with nondeterministic outcomes.

Table 10.1 compares the type of mathematical objects used in classical and quantum models for the real world. In physics, classical models describe the motion of particles using trajectories $\vec{r}(t)$, whereas quantum models use wave functions $|\psi(\vec{r},t)\rangle$. In computer science, classical information is stored in bits $i \in \{0,1\}$, whereas quantum information is stored in qubits $|x\rangle \in \mathbb{C}^2$.

Real world:

- The motion of a ball thrown in the air
- The motion of an electron through space
- The path of light particles moving through optical circuits
- The electric current flowing though a superconducting loop

Classical models:

- $x(t) \in \{\mathbb{R} \to \mathbb{R}\}$
- $\vec{r}(t) \in \{\mathbb{R} \to \mathbb{R}^3\}$
- $i \in \mathbb{Z}_2 \equiv \{0,1\}$ (a bit)
- $j \in \mathbb{Z}_d$

Quantum models:

- $|\psi(x,t)\rangle \in \{\mathbb{R} \times \mathbb{R} \to \mathbb{C}\}$
- $|\psi(\vec{r},t)\rangle \in \{\mathbb{R}^3 \times \mathbb{R} \to \mathbb{C}\}$
- $|x\rangle \in \mathbb{C}^2$ (a qubit)
- $|y\rangle \in \mathbb{C}^d$

Table 10.1: The real world and examples of classical and quantum mathematical models used to describe real-world phenomena.

Example Let's analyze the difference between classical and quantum models of the real world using an example. Consider a photon (a particle of light) going through an optical circuit that consists of several lenses, mirrors, and other optical instruments. A photon detector is placed at position x_f at the end of the circuit and the objective of the experiment is to predict if the photon will arrive at the detector and cause it to "click." The two possible outcomes of the experiment are `click` (photon arrives at detector) or `noclick` (photon doesn't arrive at detector).[1]

A classical model of the motion of the photon calculates the photon's position at all times $x(t)$ and leads to the prediction $i = 1$ (`click`) if $x_f = x(t)$, for some t. On the other hand, if the detector does not lie on the photon's trajectory, then the classical model will predict $i = 0$ (`noclick`).

A quantum model would describe the photon's trajectory through

[1] We're assuming the detector has 100% efficiency (detects every photon that arrives on it) and a zero noise (no false-positive clicks).

the circuit as a linear combination of two different possible paths:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{where} \quad |\alpha|^2 + |\beta|^2 = 1.$$

Here $|1\rangle$ describes paths that arrive at the detector, and $|0\rangle$ describes paths that don't. The coefficients α and β describe the relative "weights" of the different paths. Using the quantum model, we can obtain a probabilistic prediction of whether the detector will click or not:

$$\Pr(\texttt{noclick}) = |\alpha|^2 \quad \text{and} \quad \Pr(\texttt{click}) = |\beta|^2.$$

Both the classical and the quantum models describe the same real-world phenomenon, and the validity of both models can be tested by comparing the predictions of the models with what happens in reality.

Note the assumptions about reality the two models make are very different. The classical model assumes the photon follows a single path through the circuit, whereas the quantum model assumes the photon can take multiple paths through the circuit. Despite the difference in the mathematical "substrate" of the models, and their fundamentally different view of reality, we can compare the two models' predictions on the same footing. We cannot say one model is more "real" than the other. The only thing that is real is the photon in the optical circuit and it doesn't care whether you use classical or quantum models to describe its path.

Quantum model peculiarities

We'll now comment on the relative "intuitiveness" of classical and quantum models and introduce the concept of *quantum measurement*, which is of central importance in quantum mechanics.

Classical models have the advantage of being more intuitively understandable than quantum models. The variables in classical models often correspond to measurable aspects of real-world systems. We can identify the position variable in a classical model with the position of a particle in the real world, as measured using measuring tape. Velocity and momentum are harder to understand intuitively, but we have some general intuition about motion and collisions from everyday life. In general, classical models can be understood more easily because it's easier for us to think about a mechanistic, clockwork-like universe, in which objects push on each other, with clearly defined cause and effect, and a clock that goes click, click, click.

Quantum models, in contrast, do not enjoy such intuitive interpretations since we cannot come directly into contact with the quantum states through any of our senses. Because of this indirect connection between the states of quantum models and their predictions about the real world, quantum models are often described as mysterious

and counter intuitive. Quantum models are harder to understand in part because they use complex vector quantities to represent systems, and complex numbers are more difficult to visualize. For example, "visualizing" the complex-valued state $|\psi\rangle$ is difficult, since you have to think about both the real part and the imaginary part of $|\psi\rangle$. Even though we can't see what $|\psi\rangle$ looks like, we can describe it using an equation, and do mathematical calculations with it. In particular, we can predict the outcomes of measurements performed in the real world, and compare them with calculations obtained from the quantum state $|\psi\rangle$.

The process of *quantum measurement* is how we map the predictions of the quantum model to classical observable quantities. A quantum measurement acts on the particle's wave function $|\psi\rangle$ to produce a classical outcome. **You can think of measurements as asking the particle questions, and the measurement outcomes as the answers to these questions.**

$$\begin{aligned} \text{What is your position?} &\quad\Leftrightarrow\quad \text{position}(|\psi\rangle) = x \in \mathbb{R} \\ \text{What is your momentum?} &\quad\Leftrightarrow\quad \text{momentum}(|\psi\rangle) = p \in \mathbb{R} \\ \text{What is your spin momentum?} &\quad\Leftrightarrow\quad \text{spin}_{\updownarrow}(|\psi\rangle) = s \in \{\uparrow, \downarrow\} \end{aligned}$$

Since measurement outcomes correspond to real-world quantities that can be measured, we can judge the merits of quantum models the same way we judge the merits of classical models—in terms of the quality of their predictions.

Chapter overview

In the next section we'll describe a table-top experiment involving lasers and polarization lenses whose outcome is difficult to explain using classical physics. The remainder of the chapter will introduce the tools necessary to explain the outcome of the experiment. We'll start by introducing a special notation for vectors that is used when describing quantum phenomena (Section 10.3). In Section 10.5 we'll formally define the "rules" of quantum mechanics, also known as the *postulates* of quantum mechanics. We'll focus on learning the "rules of the game" using the simplest possible quantum systems (qubits), and define how quantum systems are prepared, how we manipulate them using *quantum operations*, and how we extract information from them using *quantum measurements*. This part of the chapter is based on the notes from the introductory lectures of a graduate-level quantum information course, so don't think you'll be getting some watered-down, hand-wavy version of quantum mechanics. You get the real stuff, because I know you can handle it. In Section 10.6 we'll use the

rules of quantum mechanics to revisit the polarization lenses experiment showing that a quantum model leads to the correct qualitative and quantitative prediction for the observed outcome. We'll close the chapter with short explanations about different applications of quantum mechanics with pointers for further exploration about each topic.

Throughout the chapter we'll focus on *matrix* quantum mechanics and use computer science language for describing quantum phenomena. Taking the computer science approach allows us to discuss the fundamental aspects of quantum theory, without the need to introduce all the physics background required to understands atoms.

Put on you safety goggle, because we're going to the lab!

10.2 Polarizing lenses experiment

We'll now describe a simple table-top experiment that illustrates the limitations of classical, deterministic reasoning. The outcomes of the experiment will highlight the need for more careful considerations of the measurements used in scientific experiments.

We'll describe the experiment using words and diagrams, but the equipment required to perform the experiment is fairly simple. You could easily reproduce the experiment in your own "lab." I encourage you to try for yourself. You'll need three polarizing lenses, a laser pointer, and three binder clips for holding the lenses upright. You can pick up polarizing lenses on the cheap from a second-hand camera shop. Any polarizing lens will do.

Background

In photography, polarizing lenses are used to filter-out certain undesirable light reflections, like reflections that occur from water surfaces or reflections from glass windows. To better understand the experiment, we need to introduce some basic notions about the physics of light, specifically, the concept of light polarization.

Light consists of photons. Photons are travelling pulses of electromagnetic energy. Electromagnetic energy can travel through space in the form of a wave. Polarization refers to the orientation of the electric field $\vec{E}$ of a propagating electromagnetic wave.

Light is normally unpolarized, meaning it corresponds to a mixture of photons whose electric and magnetic components have random orientations. A light beam is *polarized* if all photon in it have the same orientation of their electric field.

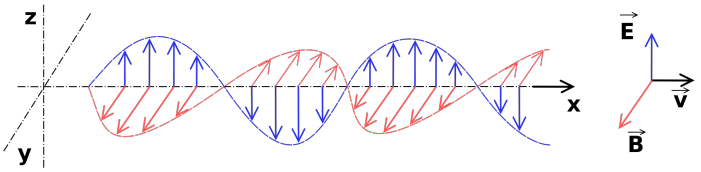


Figure 10.1: A photon is a pulse of electromagnetic energy. The energy of a photon travels in the form of a wave that has an electric component $\vec{E}$ and a magnetic component $\vec{B}$. The figure shows a photon travelling in the positive x-direction whose electric component is along the z-axis.

Light reflected from flat surfaces like the surface of a lake or a glass window becomes polarized, which means the electric components of all the reflected photons become aligned.

Photographers use this fact to selectively filter out light with a particular polarization. A *polarizing filter* or *polarizing lens* has a special coating which conducts electricity in one direction, but not in the other. You can think of the surface of a polarizing filter as being covered by tiny conducing bands that interact with the electric component of incoming light particles. Light rays that hit the filter will either pass through or be reflected depending on their polarization. Light particles whose polarization is perpendicular to the conducting lines will pass through, while light particles with polarization parallel to the conducting lines are reflected. This is because the surface of the filter has different conduction properties in different directions.

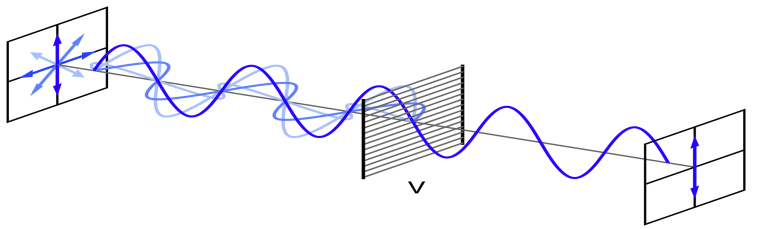


Figure 10.2: The horizontal conducting bands of a polarizing filter interact with the horizontal component of the electric field of the incoming photons and reflect them. Vertically-polarized photons pass straight through the filter, because the conducing bands are perpendicular to their electric field. Thus, a vertically polarizing filter denoted V allows only vertically polarized light to pass through.

Consider the illustration in Figure 10.3. The effect of using a vertically polarizing lens on a beam of light is to only allow vertically polarized light to pass through.

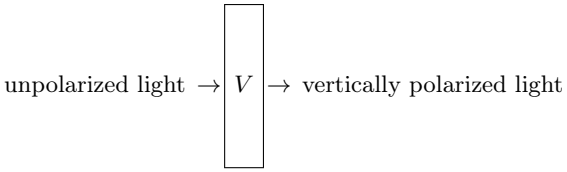


Figure 10.3: A vertically polarizing lens (V) allows only vertically polarized light particles to pass through.

In Figure 10.4 we see another aspect of using polarizing lenses. If the light is already vertically polarized, adding a second vertically polarizing lens will not affect the beam. All light that passed through the first lens will also pass through the second.

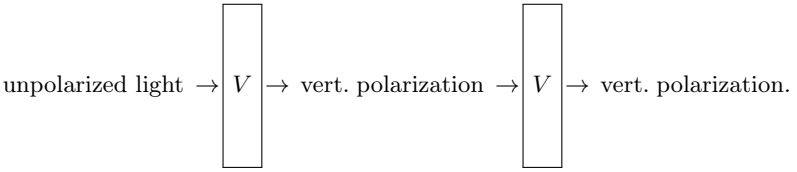


Figure 10.4: A second vertically polarizing lens has no further effect since light is already vertically polarized.

Taking a vertically polarizing lens and rotating it by 90 degrees turns it into a horizontally polarizing lens. See Figure 10.5.

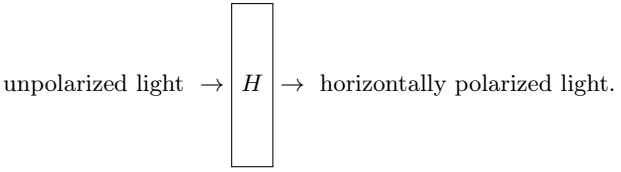


Figure 10.5: A horizontally polarizing lens (H) allows only horizontally polarized light particles to pass through.

Note that horizontally polarizing lenses and vertically polarizing lenses are complementary. Vertically polarized light will not pass through a horizontally polarizing lens. This situation is illustrated in Figure 10.6.

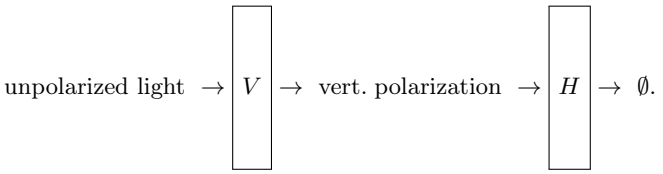


Figure 10.6: Placing a horizontally polarizing lens after the vertically polarizing lens has the effect of filtering all light. Zero photons make it through both filters, which we indicate with the empty set symbol $\emptyset$.

The above examples are meant to get you familiar with the properties of polarizing lenses, in case you don't have some to play with. If you have polarizing lenses at your disposal, you could try shining the laser pointer through them and observe when light goes through and when it gets filtered out. Use the paper clips to position the lenses on a flat surface and try to reproduce the setup in Figure 10.4. Don't worry about finding the exact orientation for "vertical." Any orientation will do so long as the first and the second polarizing lens have the same polarization. Next, you can rotate the second lens by 90° to obtain the setup shown in Figure 10.6, where the second lens has the complimentary orientation and thus rejects all light.

Example Polarized sunglasses are another example where light polarization properties can be put to good use. When a beam of light bounces from the surface of a lake, it becomes horizontally polarized. This polarization effect is due to the interaction of light's electric field at the surface of the water. A person wearing vertically polarizing lenses (polarized sunglasses) will not be able to see the sun's reflection from the water surface because the V-polarizing lenses will filter out the horizontally polarized reflection coming from the surface of the lake. This is a useful effect for people who are often out on the water, as it reduces the blinding effect of the sun's reflection.

Classical physics paradigm

Before we describe the outcome of the polarizing lenses experiment, let's take a moment to describe the assumptions about the world that 19^{th} century physicists held. Understanding this classical world view will explain why the outcomes of the experiment you're about to see are so surprising.

The classical laws of physics are deterministic, meaning they do not allow randomness. The outcomes of experiments depend on *definite* variables like properties of the particles in the experiment. It is

assumed that you can predict the outcome of any experiment given the full knowledge of the properties of particles, and if some outcome cannot be predicted, it's because you didn't know the value of some property of the particles. Everything happens for a reason, in other words. Another key assumption that a classical physicist would make is that the photon's properties are immutable, meaning we cannot change them. Classical physicists assume their experiments correspond to *passive* observations that don't affect the system's properties.

The outcome of a polarizing lens experiment is whether photons pass through a lens or get filtered. For a 19^{th} century physicists, the outcomes of such experiments depend on the polarization property of photons. You can think of each photon as carrying a tag "`H`" or "`V`" that describes its polarization. In the setup shown in Figure 10.3, each photon that makes it through the lens must have `tag="V"`, because we know by definition that a V-polarizing lens only allows vertically polarized photons to pass through. Readers familiar with SQL syntax will recognize the action of the vertically polarizing lens as the following query:

```
SELECT photon FROM photons WHERE tag="V";
```

In other words, from all the photons incident on the lens, only the vertically polarized ones are allowed to pass through. Similarly, for the H-polarizing lens shown in Figure 10.5, the filtering process can be understood as the query:

```
SELECT photon FROM photons WHERE tag="H";
```

In both cases, classical physicists would assume that whether a photon passes through a lens or not is predetermined, and it depends only on the photon's tag.

For the purpose of the discussion here, we'll restrict our attention to photons that either have horizontal (`tag="H"`) or vertical (`tag="V"`) polarization. There are other possible polarization directions, but we want to focus on the tags "`H`" and "`V`" because we know they are mutually exclusive. If a photon is horizontally polarized, then we know it will be rejected by a vertically polarizing lens. Another thing we can assert is that all photons that passed through an H-polarizing lens are *definitely not* vertically polarized, otherwise they would have been reflected and not made it through.

Polarizing lenses experiment

The physics experiment we'll describe consists of sending photons through different combinations of polarizing lenses and observing how

many of them make it through the optical circuit. We'll describe the number of photons that reach any point in the circuit qualitatively, by referring to the *optical power* reaching that point, denoted P. We keep track of the power that remains after passing through the filters, and we choose to call $P = 1$ (full brightness) the intensity of the beam after the first polarizing lens. You can think of optical power as telling you how bright of a spot is visible if you insert a piece of paper at that location. When power $P = 1$, the spot is fully bright. If $P = 0.5$ the spot is half as bright as when $P = 1$. The case $P = 0$ corresponds to zero brightness and occurs when no photons are hitting the piece of paper at all.

The starting point for the polarization experiment is an H-polarizing lens followed by a V-polarizing lens, as shown in Figure 10.7.

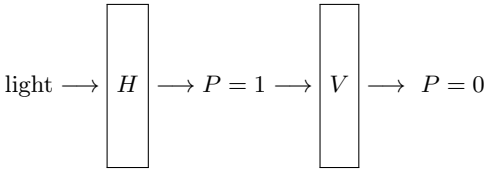


Figure 10.7: The initial setup for the polarizing lenses experiment consists of an H-polarizing lens followed by a V-polarizing lens. Only photons that have `tag="H"` can pass through the first lens, so we know no photons with `tag="V"` made it through the first lens. No photons can make it through both lenses, since the V-polarizing lens accepts only photons with `tag="V"`.

We know the photons that make it through the first filter are horizontally polarized, so it is not surprising to see that when this light hits the second lens, none of the photons make it through, since a V-polarizing lens will rejects H-polarized photons.

Adding a third lens We now introduce a third lens in between the first two, choosing the orientation of the middle lens to be different from the other two, for example in the diagonal direction. The result is shown in Figure 10.8. Suddenly light appears at the end of the circuit! Wait, what just happened here? You tell me if this is crazy or not. Intuitively, adding more filtering can only reduce the amount of light that makes it through, yet the amount of light that makes it through the circuit increases when we add the middle filter. How could adding more filtering increase light intensity? What is going on?

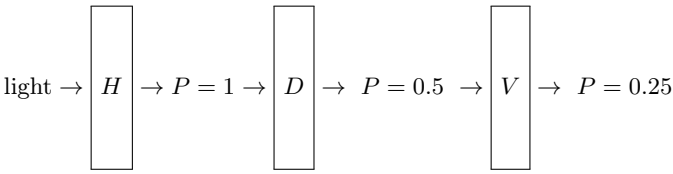


Figure 10.8: Adding an additional polarizing filter in the middle of the setup causes light to appear at the end of the optical circuit.

We pick the middle lens to be a diagonally polarizing lens D. A diagonally polarizing lens is obtained by rotating any polarizing lens by 45°. The exact choice for the middle lens is not crucial for the experiment to work; so long as it's different from the H- and V-polarizing lenses that surround it.

Classical analysis

The experimental observations illustrated in Figure 10.8 are difficult to explain using the classical way of thinking of particle properties as immutable tags, and experiments as passive observations.

Let's look at the evidence showing a particle's state can change during measurements. We'll trace the path of the photons through the optical circuit in Figure 10.8, keeping track of what we know about them at each stage. All photons that passed through the first H-polarizing lens are known to be horizontally polarized (`tag="H"`). We're sure no V-polarized photons made it through, because an H-polarizing lens is guaranteed to reject all V-polarized photons coming to it. Yet, a little bit later (after passing through the D lens) the same photons, when asked the question "are you vertical?" at the third lens, answer the question affirmatively. What kind of tagging is this? Something must be wrong with the tagging system.

It seems the photons' tag states are affected by the measurements performed on them. This fact is difficult to explain for classical physicists since they assume measurements correspond to passive observations of the systems. In the classical paradigm, measuring a photon's D-polarization using the middle lens should not affect its "`H`" and "`V`" tags.

In Section 10.5 we'll revisit this experiment after having learning the postulates of quantum mechanics. We'll see that vector-like state for describing the photon's polarization explain well the outcome of the polarizing lenses experiment and is even predicts the final light intensity of $P = 0.25$, which is observed. Before we discuss the postulates

of quantum mechanics (Section 10.5), we'll need to introduce some new notation for describing quantum states.

10.3 Dirac notation for vectors

The Dirac notation for vectors $|v\rangle$ is an alternative to the usual notations for vectors like $\vec{v}$ and $\mathbf{v}$. This section is a quick primer on Dirac notation, which is a precise and concise language for talking about vectors and matrices. This new notation will look a little weird initially, but once you get the hang of it, I guarantee you'll like it. Learning Dirac notation serves as a very good review, since we'll revisit essential linear algebra concepts like bases, vectors, inner products, and matrices. Understanding Dirac notation is essential if you're interested in learning quantum physics.

We'll now discuss several vector topics that you're familiar with, comparing standard vector notation $\vec{v}$ with the equivalent Dirac notation $|v\rangle$.

The standard basis

Consider a d-dimensional complex vector space $\mathbb{C}^d$. We refer to complex vector spaces as Hilbert spaces, in honour of David Hilbert, who did a *lot* of good things in math and physics.

Of central importance for understanding any vector space is to construct a basis for the space. A natural choice for a basis is the standard basis, which we'll denote $\{|0\rangle, |1\rangle, |2\rangle, \ldots, |d-1\rangle\}$. The basis elements are defined as follows:

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad |d-1\rangle \equiv \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Note the indices are shifted by one so the first basis element has index 0, not index 1. This zero-based indexing is chosen to make certain links between quantum theory and computer science more apparent.

The benefits of Dirac notation is that it doesn't require writing subscripts. To refer to a vector associated with properties a, b, and c, we can write $|a, b, c\rangle$, instead of the more convoluted expression $\vec{v}_{a,b,c}$.

We'll now restrict attention to the two-dimensional complex vector space $\mathbb{C}^2$. The results and definitions presented below apply equally well to vectors in $\mathbb{C}^d$ as to vectors in $\mathbb{C}^2$.

Vectors

In Dirac notation, a vector in $\mathbb{C}^2$ is denoted as a *ket*:

$$|v\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \Leftrightarrow \qquad \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

where $\alpha \in \mathbb{C}$ and $\beta \in \mathbb{C}$ are the *coefficients* of $|v\rangle$ and $\{|0\rangle, |1\rangle\}$ is the standard basis for $\mathbb{C}^2$:

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Why do we call the angle-bracket thing a "ket" you ask? Let me tell you about the *bra*, and then it will start to make sense.

The Hermitian transpose of the ket-vector $|v\rangle = \alpha|0\rangle + \beta|1\rangle$ is the *bra*-vector $\langle v|$:

$$\langle v| = \overline{\alpha}\langle 0| + \overline{\beta}\langle 1| \qquad \Leftrightarrow \qquad [\overline{\alpha}, \overline{\beta}] = \overline{\alpha}[1, 0] + \overline{\beta}[0, 1].$$

Recall that the Hermitian transpose (complex transpose) is the combination of the regular transpose ($\vec{v} \to \vec{v}^{\mathsf{T}}$) and the complex conjugation of each entry in the vector ($v_i \to \overline{v_i}$), and is denoted as the dagger operator ($\dagger$). Observe how much simpler the *bra* notation for the Hermitian transpose of a vector is compared to the other notations we've seen thus far:

$$\langle v| \qquad \Leftrightarrow \qquad \vec{v}^{\dagger} \equiv \overline{(\vec{v}^{\mathsf{T}})} \equiv (\overline{\vec{v}})^{\mathsf{T}}.$$

But why call $\langle v|$ a "bra"? Is this some sort of sexist joke?

What happens when you put a bra $\langle v|$ next to a ket $|w\rangle$? A braket $\langle v|w\rangle$, and a braket that looks very similar to the brackets used to denote the inner product between two vectors. Observe how easy it is to calculate the inner product between the vectors $|v\rangle = v_0|0\rangle + v_1|1\rangle$ and $|w\rangle = w_0|0\rangle + w_1|1\rangle$ in Dirac notation:

$$\langle v|w\rangle = \overline{v_0}w_0 + \overline{v_1}w_1 \qquad \Leftrightarrow \qquad \vec{v}^{\dagger}\vec{w} \equiv \overline{\vec{v}} \cdot \vec{w}.$$

Complex conjugation was already applied to the coefficients of $|v\rangle$ when transforming it into a bra-vector $\langle v|$, thus we can simply "put together" the bra and the ket to compute the inner product. The bra notation $\langle v|$ contains the Hermitian transpose so it removes the need for the dagger symbol. This trivial simplification of the notation for inner product turns out to be very useful. Inner products are the workhorse for calculating vector coefficients, finding projections, performing change-of-basis transformations, and in countless applications. The simpler notation for inner products will enable us to dig deeper into each of these aspects of linear algebra, without getting overwhelmed by notational complexity.

The inner product of $|v\rangle = \alpha|0\rangle + \beta|1\rangle$ with itself is

$$\begin{aligned}\langle v|v\rangle &= \big(\overline{\alpha}\langle 0| + \overline{\beta}\langle 1|, \alpha|0\rangle + \beta|1\rangle\big) \\ &= \overline{\alpha}\alpha\langle 0|0\rangle + \overline{\alpha}\beta\underbrace{\langle 0|1\rangle}_{0} + \overline{\beta}\alpha\underbrace{\langle 1|0\rangle}_{0} + \overline{\beta}\beta\langle 1|1\rangle \\ &= |\alpha|^2 + |\beta|^2.\end{aligned}$$

The ability to express vectors, inner products, and vector coefficients in a concise and consistent manner is why everyone likes Dirac notation. Dear readers, are you still with me? You can't tell me what we've seen so far is too complicated, right? Let's continue then. The main virtue of Dirac's bra-ket notation is that it is sufficiently simple as to be used in equations without the need to define new variables v_i for vector coefficients. We'll look at this more closely in the next section.

Vector coefficients

The *coefficients* v_i of a vector $\vec{v}$ with respect to an orthonormal basis $\{\hat{e}_i\}$ are computed using the inner product $v_i \equiv \hat{e}_i^\dagger \vec{v}$. In Dirac notation, the coefficients of $|v\rangle$ with respect to the standard basis $\{|0\rangle, |1\rangle\}$ can be written as $\langle i|v\rangle$. We can write any vector $|v\rangle$ as a linear combination of kets, with bra-kets as coefficients:

$$|v\rangle = \underbrace{\langle 0|v\rangle}_{v_0}|0\rangle + \underbrace{\langle 1|v\rangle}_{v_1}|1\rangle.$$

The expression $\langle i|v\rangle$ is simple enough and it explicitly defines the i^{th} coefficient of $|v\rangle$. We therefore don't need to define the variable v_i.

Another basis for the vector space $\mathbb{C}^2$ is the *Hadamard basis* which corresponds to the standard basis rotated by 45° in the counterclockwise direction:

$$\begin{aligned}|+\rangle &\equiv \tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle, \\ |-\rangle &\equiv \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle.\end{aligned}$$

The Hadamard basis, henceforth denoted $B_h \equiv \{|+\rangle, |-\rangle\}$, is an orthonormal basis:

$$\langle +|+\rangle = 1, \quad \langle +|-\rangle = 0, \quad \langle -|+\rangle = 0, \quad \langle -|-\rangle = 1.$$

Since the Hadamard basis B_h is an orthonormal basis, we can express any vector $|v\rangle \in \mathbb{C}^2$ as a linear combination of kets:

$$|v\rangle = \underbrace{\langle +|v\rangle}_{v_+}|+\rangle + \underbrace{\langle -|v\rangle}_{v_-}|-\rangle.$$

Note the coefficients of the linear combination are computed using the inner product with the corresponding basis vector $\langle +|v\rangle$ and $\langle -|v\rangle$. The bra-ket notation allows us to refer to the coefficients of $|v\rangle$ with respect to $\{|+\rangle, |-\rangle\}$ without the need to define variables v_+ and v_-.

Vector coefficients with respect to different bases are often used in calculations. Using the usual vector notation, we must specify which basis is being used as a subscript. For example, the same vector $\vec{v}$ can be expressed as a coefficient vector $\vec{v} = (v_0, v_1)^{\mathsf{T}}_{B_s}$ with respect to the standard basis B_s, or as a coefficient vector $\vec{v} = (v_+, v_-)^{\mathsf{T}}_{B_h}$ with respect to the Hadamard basis. In the bra-ket notation, the coefficients with respect to B_s are $\langle 0|v\rangle$ and $\langle 1|v\rangle$, and the coefficients with respect to B_h are $\langle +|v\rangle$ and $\langle -|v\rangle$, making the choice of basis evident.

Change of basis

Consider the task of finding the *change-of-basis* matrix ${}_{B_h}[\mathbb{1}]_{B_s}$ that converts vectors from the standard basis $B_s = \{(1,0),(0,1)\}$, to the Hadamard basis $B_h = \{(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}), (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})\}$.

Using the standard approach for finding change-of-basis matrices discussed in Section 5.3 (page 198), we know the columns of ${}_{B_h}[\mathbb{1}]_{B_s}$ contains the coefficients of $(1,0)$ and $(0,1)$ as expressed with respect the basis B_h:

$$ {}_{B_h}[\mathbb{1}]_{B_s} = {}_{B_h}\begin{bmatrix} \langle +|0\rangle & \langle +|1\rangle \\ \langle -|0\rangle & \langle -|1\rangle \end{bmatrix}_{B_s} = {}_{B_h}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}_{B_s}. $$

We can now compute the coefficients of any vector $\vec{v} \equiv (v_0, v_1)^{\mathsf{T}}_{B_s}$ with respect to the Hadamard basis, by multiplying $(v_0, v_1)^{\mathsf{T}}_{B_s}$ by the change-of-basis matrix:

$$ \begin{aligned} \begin{bmatrix} v_+ \\ v_- \end{bmatrix}_{B_h} &= {}_{B_h}[\mathbb{1}]_{B_s} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}_{B_s} \\ &= {}_{B_h}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}_{B_s} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}_{B_s} \\ &= \begin{bmatrix} \frac{1}{\sqrt{2}}(v_0 + v_1) \\ \frac{1}{\sqrt{2}}(v_0 - v_1) \end{bmatrix}_{B_h}. \end{aligned} $$

This is the usual approach for computing the coefficients of a vector in one basis in terms of the coefficients of another basis using matrix notation.

Consider now the same change-of-basis operation, but with calculations carried out in Dirac notation. Given the vector $\vec{v} = (v_0, v_1)_{B_s} =$

$v_0|0\rangle + v_1|1\rangle$ expressed as coefficients with respect to the standard basis B_s, we want to find $\vec{v} = (v_+, v_-)_{B_h} = v_+|+\rangle + v_-|-\rangle$. Starting from the definition of v_+ and v_-, we obtain

$$\begin{aligned}
\vec{v} &\equiv (v_+, v_-)^{\mathsf{T}}_{B_h} \\
&\equiv \langle +|v\rangle|+\rangle + \langle -|v\rangle|-\rangle \\
&= \langle +|\big(v_0|0\rangle + v_1|1\rangle\big)|+\rangle + \langle -|\big(v_0|0\rangle + v_1|1\rangle\big)|-\rangle \\
&= \big(v_0\langle +|0\rangle + v_1\langle +|1\rangle\big)|+\rangle + \big(v_0\langle -|0\rangle + v_1\langle -|1\rangle\big)|-\rangle \\
&= \underbrace{\tfrac{1}{\sqrt{2}}\big(v_0 + v_1\big)}_{v_+}|+\rangle + \underbrace{\tfrac{1}{\sqrt{2}}\big(v_0 - v_1\big)}_{v_-}|-\rangle.
\end{aligned}$$

Working from the definitions of vectors and their components and using only basic algebra rules we can perform the change-of-basis operation, without explicitly constructing the change-of-basis matrix.

Outer products

Recall the *outer product* operation for vectors that we introduced in Section 6.2 (see page 237). The expression $|0\rangle\langle 0|$ is equivalent to the *projection* onto the subspace spanned by the vector $|0\rangle$:

$$|0\rangle\langle 0| \qquad \Leftrightarrow \qquad \begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}.$$

We can verify this by considering the product of $|0\rangle\langle 0|$ and an arbitrary vector $|v\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$|0\rangle\langle 0|\,(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle\underbrace{\langle 0|0\rangle}_{1} + \beta|0\rangle\underbrace{\langle 0|1\rangle}_{0} = \alpha|0\rangle.$$

The ability to easily express outer products is another win for Dirac notation. For example, the projection onto the $|+\rangle$-subspace is $|+\rangle\langle +|$.

Matrices

Now get ready for some crazy stuff. It turns out outer-product expressions are useful not only for projections, but can in fact represent any matrix. Consider the linear operator $A : \mathbb{C}^2 \to \mathbb{C}^2$ and its matrix representation with respect to the standard basis:

$${}_{B_s}[A]_{B_s} \equiv \begin{bmatrix}a_{00} & a_{01}\\a_{10} & a_{11}\end{bmatrix}_{B_s}.$$

Instead of positioning the coefficients in an array, we can represent A as a linear combination of outer-products:

$$A \;\equiv\; a_{00}|0\rangle\langle 0| + a_{01}|0\rangle\langle 1| + a_{10}|1\rangle\langle 0| + a_{11}|1\rangle\langle 1|.$$

The coefficient a_{10} describes the multiplier A applies between the coefficient of $|0\rangle$ in its input and the coefficient of $|1\rangle$ in its output. The expression $a_{10}|1\rangle\langle 0|$ is a concise description of the same story. The $\langle 0|$ in this expression will "select" only the $|0\rangle$ coefficient of the input, and the $|1\rangle$ indicates this term contributes to the $|1\rangle$ component of the output.

The coefficients of the matrix representation ${}_{B_s}[A]_{B_s}$ depend on the choice of bases for the input and output spaces. The value of the coefficient a_{ij} in the matrix representation is computed by "probing" the matrix with the j^{th} basis element of the input basis, and observing the value of the i^{th} element in the resulting output. We can express the entire "probing procedure" easily in Dirac notation: $a_{ij} \equiv \langle i|A|j\rangle$. Thus, we can write the matrix coefficients as follows:

$$ {}_{B_s}\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}_{B_s} = {}_{B_s}\begin{bmatrix} \langle 0|A|0\rangle & \langle 0|A|1\rangle \\ \langle 1|A|0\rangle & \langle 1|A|1\rangle \end{bmatrix}_{B_s} $$

In fact, we don't need matrix notation or the coefficients a_{ij} at all. Instead, we can express A as a linear combination of outer products, with appropriately chosen coefficients:

$$ A \equiv \langle 0|A|0\rangle|0\rangle\langle 0| + \langle 0|A|1\rangle|0\rangle\langle 1| + \langle 1|A|0\rangle|1\rangle\langle 0| + \langle 1|A|1\rangle|1\rangle\langle 1|. $$

Let's verify the formula for the $a_{10} = \langle 1|A|0\rangle$-coefficient of A, to see how this linear combination of outer products thing works. We start from the definition $A \equiv a_{00}|0\rangle\langle 0| + a_{01}|0\rangle\langle 1| + a_{10}|1\rangle\langle 0| + a_{11}|1\rangle\langle 1|$, and multiply A by $|0\rangle$ from the right, and by $\langle 1|$ from the left:

$$ \begin{aligned} \langle 1|A|0\rangle &= \langle 1|\Big(a_{00}|0\rangle\langle 0| + a_{01}|0\rangle\langle 1| + a_{10}|1\rangle\langle 0| + a_{11}|1\rangle\langle 1|\Big)|0\rangle \\ &= \langle 1|\Big(a_{00}|0\rangle \underbrace{\langle 0|0\rangle}_{1} + a_{01}|0\rangle \underbrace{\langle 1|0\rangle}_{0} + a_{10}|1\rangle \underbrace{\langle 0|0\rangle}_{1} + a_{11}|1\rangle \underbrace{\langle 1|0\rangle}_{0}\Big) \\ &= \langle 1|\Big(a_{00}|0\rangle + a_{10}|1\rangle\Big) \\ &= a_{00}\underbrace{\langle 1|0\rangle}_{0} + a_{10}\underbrace{\langle 1|1\rangle}_{1} \\ &= a_{10}. \end{aligned} $$

So indeed $\langle 1|A|0\rangle$ is the same as a_{10}. In fact, we'll rarely use coefficient notation a_{10}, since $\langle 1|A|0\rangle$ is just as easy to write, and much more intuitive: the a_{10}-coefficient of A is what you obtain when you "sandwich" the matrix A between the vectors $\langle 1|$ on the left and $|0\rangle$ on the right.

In Dirac notation, the basis appears explicitly in expressions for coefficients of a matrix. We can define the coefficients of A in any

other basis very easily and precisely. The representation of A with respect to the Hadamard basis $B_h = \{|+\rangle, |-\rangle\}$ is

$$ {}_{B_h}[A]_{B_h} \equiv {}_{B_h}\begin{bmatrix} \langle +|A|+\rangle & \langle +|A|-\rangle \\ \langle -|A|+\rangle & \langle -|A|-\rangle \end{bmatrix}_{B_h}, $$

or equivalently

$$ A = \langle +|A|+\rangle |+\rangle\langle +| + \langle +|A|-\rangle |+\rangle\langle -| + \langle -|A|+\rangle |-\rangle\langle +| + \langle -|A|-\rangle |-\rangle\langle -|. $$

The coefficient $\langle +|A|+\rangle$ is the a_{++}-coefficient of the matrix representation of A with respect to the Hadamard basis B_h.

Summary Dirac notation is a convenient way to represent linear algebra concepts: vectors $|v\rangle$ and their Hermitian conjugates $\langle v|$, vector coefficients $\langle i|v\rangle$, inner products $\langle v|w\rangle$, outer products $|v\rangle\langle w|$, and matrix coefficients $\langle i|A|j\rangle$. Because of this expressiveness, Dirac notation is widely used in modern chemistry, physics, and computer science, when discussing quantum mechanical topics. In particular, Dirac notation is a core component in the study of quantum physics. In fact, we could say that if you understand Dirac notation, you understand half of quantum mechanics already.

Exercises

E10.1 Consider the ket vector $|u\rangle = \alpha|0\rangle + \beta|3\rangle$ and the the bra vector $\langle v|$, where $|v\rangle = a|1\rangle + b|2\rangle$. Express $|u\rangle$ and $\langle v|$ as four dimensional vector of coefficients. Indicate whether your answers correspond to row vectors or column vectors.

E10.2 Express the vector $\vec{w} = (1, i, -1)^{\mathsf{T}}_{B_s} \in \mathbb{C}^3$ as a ket $|w\rangle$ and as a bra $\langle w|$. Compute its length $\|\vec{w}\| = \sqrt{\langle w|w\rangle}$.

E10.3 Find the determinant of the following matrix:

$$ A = 1|0\rangle\langle 0| + 2|0\rangle\langle 1| + 3|1\rangle\langle 0| + 4|1\rangle\langle 1|. $$

10.4 Quantum information processing

Digital technology is sought after because of the computational, storage, and communication advantages that come with manipulating digital information instead of continuous-time signals. Similarly, quantum technology is interesting because of the computational and communication applications it enables. The purpose of this section is to equip you with a mental model for thinking about quantum information processing tasks, based on an analogy with digital information processing pipelines, which may be more familiar to you.

The use of quantum technology for information processing tasks is no more mysterious than the use of digital technology for information processing tasks. Playing a digital song recording on your `mp3` player involves a number of processing, conversion, and signal amplification steps. Similarly, using a quantum computer involves several conversion, processing, and measurement steps. In both cases you input some data into a machine, wait for the machine to process the data, and then read out the answer. The use of digital and quantum technology can both be described operationally as black box processes, whose internals are not directly accessible to us. In both cases the intermediate representation of the data is in a format that is unintelligible: we can't "see" quantum states, but we can't "see" digital information either. Think about it, is "seeing" the raw ones and zeros of an `mp3` file really all that helpful? Can you tell which artist plays the song 010100101010101000111...? To understand information processing in the digital and quantum worlds, we must study the adaptors for converting between the internal representations and world we can see.

In order to highlight the parallel structure between the digital information processing and quantum information processing, we'll now look at an example of a digital information processing task.

Digital signal processing

A *sound card* is a computer component that converts between analog signals that we can hear and digital signals that computers understand. Sound is "digitized" using an analog-to-digital converter (ADC). For music playback we use a digital-to-analog converter (DAC), which transforms digital sounds into analog sound vibrations to be reproduced by the speakers. The ADC corresponds to the microphone jack on the sound card; the DAC output of the sound card goes to the line-out and headphone jacks.

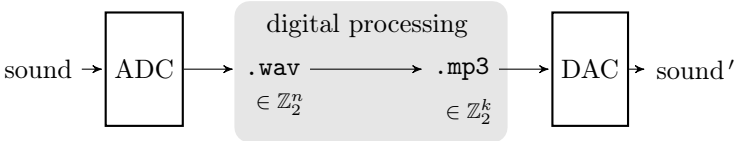


Figure 10.9: A digital information processing pipeline for sound recording and playback. Sound vibrations are captured by a microphone and converted to digital form using an analog-to-digital converter (ADC). Next the digital `wav` file is converted to the more compact `mp3` format using digital processing. In the last step, sound is converted back into analog sound vibrations by a digital-to-analog converter (DAC).

Figure 10.9 illustrates a digital information processing pipeline for sound. We use an *analog-to-digital converter* (ADC) to transform the analog sound into digital form, we then process it in the digital world, and finally use a *digital-to-analog converter* (DAC) to transform from the digital world back to the analog world. If we're successful in the music encoding, processing, and playback steps, the final output will sound like the original sound recorded. The grey-shaded region in the figure corresponds to the *digital world*. We've chosen to show only one, simple information processing task: the compression of a large `wav` file ($n = 50$[MB]) into a smaller `mp3` file ($k = 5$[MB]), which we previously discussed in Section 8.11 (page 385).

The example of `mp3` compression is just one many uses of digital processing that are possible once we convert information into digital form. Digital information is *very* versatile, you can compress it, encrypt it, encode it, transcode it, store it, and—most important of all—transmit it over the Internet.

Quantum information processing

Quantum processing pipelines are analogous to digital information processing pipelines. The process of *state preparation* corresponds to the analog-to-digital conversion step. Quantum *measurement* corresponds to the digital-to-analog conversion step. We use state preparation to put information into a quantum computer, and measurement to read information out.

Figure 10.10 shows an example of a quantum information processing pipeline. A classical bitstring $x \in \mathbb{Z}_2^k$ is supplied, and after state preparation, quantum processing, and measurement steps, the classical bitstring $y \in \mathbb{Z}_2^\ell$ is produced. The crucial difference with the classical information processing, is that quantum information processing is defined as information processing with quantum states. Thus, quantum computation is more general than a classical computation. Instead of working with *bits* and using functions $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^\ell$, we're working with *qubits* and use *quantum operations* $T : \mathbb{C}^m \to \mathbb{C}^n$ to perform computations.

The other big difference between classical and quantum computation, is that you can read out the output state $|y\rangle$ only once. You can think of **quantum measurement as asking a question** about the state $|y\rangle$. You're free to choose to perform any measurement, but **you can ask only one question**, since quantum measurement disrupt the state of a system. Also the answer you obtain to your question depends *probabilistically* on the state $|y\rangle$.

In the next section, we'll discuss the components of the quantum information processing pipeline in more details. We'll introduce the

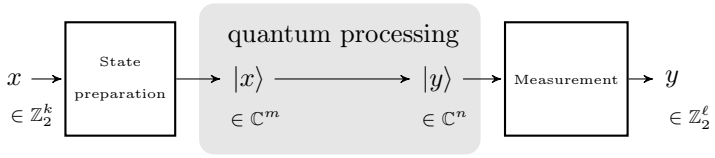


Figure 10.10: A quantum information processing pipeline. A classical bitstring x of length k is used as instructions for preparing an m-dimensional quantum state $|x\rangle$. Next, quantum operations are performed on the state $|x\rangle$ to convert it to the output state $|y\rangle$. Finally, the state $|y\rangle$ is *measured* in order to obtain the classical bitstring y as output.

four *postulates* of quantum mechanics, which specify how quantum systems are represented and what we can do with them. The postulates of quantum mechanics roughly correspond to the conversion steps illustrated in Figure 10.10. One postulate defines how quantum state are prepared, another postulate describe the types operations we can perform on quantum states, and a third postulated formally defines the process of quantum measurement. The next section is the "explanation of quantum mechanics in the space on the back of an envelope," alluded to in the introduction. We've set the scene, introduced Dirac notation, so we can finally go into the good stuff.

10.5 Postulates of quantum mechanics

The *postulates* of quantum mechanics dictate the rules for working with the "quantum world." The four postulates of quantum mechanics define

- What quantum states are
- Which quantum operations are allowed on quantum states
- How to extract information from quantum systems by carrying out measurements on them
- How to represent composite quantum systems

The postulates of quantum mechanics specify the structure that all quantum theories must have, and are common to all fields that use quantum mechanics: physics, chemistry, engineering, and quantum information. Note the postulates are not provable or derivable from a more basic theory: scientists simply take the postulates as facts and make sure their theories embody these principles.

Quantum states

Quantum states are modelled as special type of vectors. The *state* of a d-dimensional quantum system is a unit-length vector $|\psi\rangle$, in a d-dimensional complex inner-product vector space $\mathbb{C}^d$, which we call Hilbert space.

Postulate 1. To every isolated quantum system is associated a complex inner product space (Hilbert space) called the *state space*. A state is described by a unit-length vector in state space.

The following additional comments apply to states that describe quantum systems:

- The requirement that state vectors must have unit length will become important when we discuss the probabilistic nature of quantum measurements.
- The *global phase* of a quantum state vector doesn't matter. Thus $|\psi\rangle \equiv e^{i\theta}|\psi\rangle, \forall\theta \in \mathbb{R}$. This means the vectors $|\psi\rangle$, $-|\psi\rangle$, and $i|\psi\rangle$ all represent the same quantum state.
- Each physical system lives in its own Hilbert space, usually denoted with the same label as the system, $|\psi\rangle_1 \in V_1$ and $|\psi\rangle_2 \in V_2$.

In general, quantum states can be vectors in d-dimensional or sometimes even infinitely-dimensional Hilbert spaces. We'll focus on two-dimensional quantum systems.

The qubit In analogy with the classical bits $\in \{0, 1\}$, we call two-dimensional quantum systems *qubits* $\in \mathbb{C}^2$, which is short for *quantum bit*. Many physical systems like the polarization of a photons and the spin of electrons can all be represented as qubits. A qubit is a unit-length vector in a two-dimensional Hilbert space $\mathbb{C}^2$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where

$$\begin{aligned} &\alpha \in \mathbb{R}, && \text{(global phase doesn't matter)} \\ &\beta \in \mathbb{C}, \\ &|\alpha|^2 + |\beta|^2 = 1. \end{aligned}$$

Recall $|0\rangle$ and $|1\rangle$ are the elements of standard basis for $\mathbb{C}^2$:

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The restriction to α being a real number follows from the fact that global phase of a quantum state can be ignored. The condition that a quantum state must have unit length is equivalent to the constraint $|\alpha|^2 + |\beta|^2 = 1$.

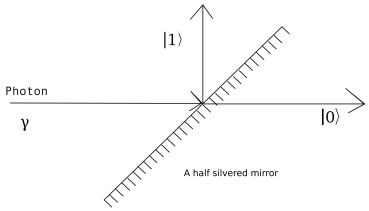


Figure 10.11: A photon encounters a half-silvered mirror. The photon can take one of the two possible paths, so we describe it as the superposition $|\gamma\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, where $|0\rangle$ describes the photon passing through the mirror, and $|1\rangle$ describes the photon being reflected.

The notion of a qubit is an abstract concept and many physical systems can embody it. In Figure 10.11, the information of a qubit is encoded in the path of the photon takes after it encounters the half-silvered mirror. Qubits are a device-independent representation of quantum information, similar to how classical bits are device-independent representations for classical information. A bit is a bit, regardless of whether it is transmitted over the network, stored in RAM, or burned to a DVD. Similarly a qubit is a qubit, regardless of whether it's encoded in the polarization of a photon, an electron's spin, or in the direction of magnetic flux of a superconducting loop.

Quantum state preparation The operation of encoding some classical information into a quantum system is called *state preparation*. Imagine an apparatus that prepares quantum systems in one of several possible quantum states, depending on the position of the "control switch" x of the machine.

$$x \longrightarrow \boxed{\text{state preparation}} \longrightarrow |x\rangle$$

Figure 10.12: The classical input x is used to prepare a quantum system. The state $|x\rangle$ produced when the input is x in one of several possible states.

An example of quantum state preparation is a machine that can produce photons in two different polarizations $|H\rangle$ and $|V\rangle$. If the input $x = 0$ is specified, the machine will produce the state $|H\rangle$, and if $x = 1$ is specified as input, the machine will produce $|V\rangle$.

Having defined quantum states, it's now time to discus what we can *do* with quantum states. What are the allowed quantum operations?

Quantum operations

The second definition that belongs on the back-of-the-enveloper explanations, is the identification of *quantum operations* with unitary transformations acting on quantum states. The following diagram shows the the quantum operation U being applied to the input state $|\psi\rangle$ to produce the output state $|\psi'\rangle$.

$$|\psi\rangle \longrightarrow \boxed{\text{U}} \longrightarrow |\psi'\rangle$$

More generally, all quantum operations can perform on quantum states are represented by unitary transformations as codified in the second postulate of quantum mechanics.

Postulate 2. Time evolution of an isolated quantum system is unitary. If the state at time t_1 is $|\psi_1\rangle$ and at time t_2 is $|\psi_2\rangle$ then $\exists$ unitary U such that $|\psi_2\rangle = U|\psi_1\rangle$.

Recall that a unitary matrix U obeys $U^\dagger U = UU^\dagger = \mathbb{1}$. The second postulate ensures that quantum states will maintain their unit-length property after quantum operation are performed on them. Assume the quantum system starts from a state $|\psi_1\rangle$ that has unit length $\||\psi_1\rangle\|^2 = \langle\psi_1|\psi_1\rangle = 1$. After the unitary U is applied, the state after evolution will be $|\psi_2\rangle = U|\psi_1\rangle$ and its norm is $\||\psi_2\rangle\|^2 = \|U|\psi_1\rangle\|^2 = \langle\psi_1|U^\dagger U|\psi_1\rangle = \langle\psi_1|\mathbb{1}|\psi_1\rangle = \langle\psi_1|\psi_1\rangle = 1$. In other words, **quantum operations are length-preserving**.

We'll now define several quantum gates that perform useful unitary operations on qubits.

Example 1: phase gate The Z operator is defined by its action on the elements of the standard basis.

$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -1|1\rangle.$$

The Z operator leaves the $|0\rangle$ unchanged but flips the phase of $|1\rangle$.

Given the knowledge of the actions of the Z operator on the vectors of the standard basis, we can construct its matrix representations:

$$Z = \underset{B_s}{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}}_{B_s} \equiv |0\rangle\langle 0| - |1\rangle\langle 1|.$$

Example 2: NOT gate The X operator is defined by the following actions on the elements of the standard basis:

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle.$$

The X operator acts as a "quantum NOT gate" changing $|0\rangle$s into $|1\rangle$s, and $|1\rangle$s into $|0\rangle$s. The matrix representation of the X operator is:

$$X = \underset{B_s}{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{B_s}} \equiv |0\rangle\langle 1| + |1\rangle\langle 0|.$$

Example 3: Hadamard gate The Hadamard operator takes the elements of the standard basis to the elements of the Hadamard basis $|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle, \qquad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle.$$

You can also think of the H operator as a 45° counter-clockwise rotation. The matrix representation of the Hadamard gate is

$$H = \underset{B_s}{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}_{B_s}}.$$

By linearity, we can deduce the effects of the operators Z, X, and H on an arbitrary qubit $\alpha|0\rangle + \beta|1\rangle$:

$$\begin{aligned} Z(\alpha|0\rangle + \beta|1\rangle) &= \alpha|0\rangle - \beta|1\rangle, \\ X(\alpha|0\rangle + \beta|1\rangle) &= \beta|0\rangle + \alpha|1\rangle, \\ H(\alpha|0\rangle + \beta|1\rangle) &= \tfrac{\alpha+\beta}{\sqrt{2}}|0\rangle + \tfrac{\alpha-\beta}{\sqrt{2}}|1\rangle. \end{aligned}$$

Example 4 The effect of the the operator XZ corresponds to the combination of the effects of the Z and X operators. We can understand the action of XZ either by applying it to a arbitrary qubit $\alpha|0\rangle + \beta|1\rangle$:

$$XZ(\alpha|0\rangle + \beta|1\rangle) = -\beta|0\rangle + \alpha|1\rangle,$$

or by multiplying together the operator's matrix representations:

$$XZ = \underset{B_s}{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{B_s}} \underset{B_s}{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}_{B_s}} = \underset{B_s}{\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}_{B_s}}.$$

In general, many possible other quantum operations and combinations of operations are possible. The ones presented above are simply the most commonly used. Note that unitary time evolution is invertible: for every quantum gate G there exists an inverse gate G^{-1} such that $G^{-1}G = \mathbb{1}$.

Exercises

E10.4 Compute $XHHY(\alpha|0\rangle + \beta|1\rangle)$.

E10.5 Compute XX, XY, and YX.

We're now two-thirds down the back of the envelope, and it's time to talk about the third fundamental idea in quantum mechanics: quantum measurements.

Quantum measurements

A quantum measurement performed on a quantum system corresponds to a collections of projection operators $\{\Pi_i\}$ that act on the Hilbert space. A measurement with n possible outcomes is represented by n projection operators $\{\Pi_1, \Pi_2, \ldots, \Pi_n\}$. The projection operators form a *decomposition of the identity*, meaning their sum is the identity matrix:

$$\sum_{i=1}^{n} \Pi_i = \mathbb{1}.$$

Intuitively, the n different projection operators correspond to n different alternatives for the evolution of the quantum system. Performing the measurement is like asking "which is it going to be?" and letting the system decide which path it wants to take. *Born's rule* is used to assign probabilities to different measurement outcomes.

Postulate 3. A quantum measurement is modelled by a collection of projection operators $\{\Pi_i\}$ that act on the state space of the system being measured and satisfy $\sum_i \Pi_i = \mathbb{1}$. The index i labels the different measurement outcomes.

The probability of outcome i when performing measurement $\{\Pi_i\}$ on a quantum system in the state $|\psi\rangle$ is given by the square-modulus of the state after the applying the i^{th} projection operator:

$$\Pr(\{\text{outcome } i \text{ given state } |\psi\rangle\}) \equiv \left\|\Pi_i|\psi\rangle\right\|^2 \qquad \text{(Born's rule)}.$$

When outcome i occurs, the post-measurement state of the system is

$$|\psi_i'\rangle \equiv \frac{\Pi_i|\psi\rangle}{\|\Pi_i|\psi\rangle\|}.$$

Let's unpack this definition to see what is going on.

Born's rule For the measurement defined by the projection operators $\{\Pi_1, \Pi_2, \ldots, \Pi_n\}$, Born's rule states the probability of outcome i is $\langle\psi|\Pi_i|\psi\rangle$.

This expression for the square of the modulus of the overlap between $|\psi\rangle$ and Π_i can be written in several equivalent ways:

$$\left\|\Pi_i|\psi\rangle\right\|^2 = (\Pi_i|\psi\rangle, \Pi_i|\psi\rangle) = \langle\psi|\Pi_i\Pi_i|\psi\rangle = \langle\psi|\Pi_i|\psi\rangle.$$

where the last equality follows from idempotence property of projectors $\Pi_i = \Pi_i^2$. The last expression with the projection operator "sandwiched" by two copies of the quantum state is the physicist's usual way of expressing Born's rule, defining $\Pr(\{\text{outcome } i\}) \equiv \langle\psi|\Pi_i|\psi\rangle$. For the class of projective measurements we're discussing here, the two definitions are equivalent.

The set of projection $\{\Pi_1, \Pi_2, \ldots, \Pi_n\}$ forms a decomposition of the identity $\mathbb{1} = \sum_i \Pi_i$. This guarantees the probability distribution of the different outcomes is well normalized:

$$\begin{aligned} 1 = \|\mathbb{1}|\psi\rangle\|^2 &= \|(\Pi_1 + \cdots + \Pi_n)|\psi\rangle\|^2 \\ &\overset{\text{py}}{=} \|\Pi_1|\psi\rangle\|^2 + \cdots + \|\Pi_n|\psi\rangle\|^2 \\ &= \Pr(\{\text{outcome } 1\}) + \cdots + \Pr(\{\text{outcome } n\}) \,. \end{aligned}$$

That's good to check, otherwise Kolmogorov would be angry with us. Note the equality labelled $\overset{\text{py}}{=}$ follows from Pythagoras' theorem; we're using the fact that operators $\{\Pi_i\}$ are mutually orthogonal.

Post-measurement state When outcome i occurs, Postulate 3 tells us the state of the quantum system becomes $|\psi_i'\rangle \equiv \frac{\Pi_i|\psi\rangle}{\|\Pi_i|\psi\rangle\|}$, which is the result of applying the projection Π_i to obtain $\Pi_i|\psi\rangle$, and then normalizing the state so $\||\psi_i'\rangle\| = 1$.

Measurements are not passive observations! Quantum measurement is an invasive procedure that typically changes the state of the system being measured. In general $|\psi_i'\rangle \neq |\psi\rangle$ and we say the state is *disturbed* by the measurement, thought it's also possible that $|\psi_i'\rangle = |\psi\rangle$ when the input state lives entirely within the image subspace of one of the projection operators: $|\psi_i'\rangle = \Pi_i|\psi\rangle = |\psi\rangle$.

Another way to describe what happens during a quantum measurement is to say the state $|\psi\rangle$ *collapses* into the state $|\psi_i'\rangle$. We won't use this the terminology of "collapse" here because this terminology often leads to magical thinking and whacky interpretations of the mechanism behind quantum measurement. Of the four postulates of quantum mechanics, Postulate 3 is the most debated one, with various scientists denying it, or attaching free-for-all interpretations

about how this "collapse" occurs. This is usually where the "human observer" aspect of pop-psychology books connects. The reasoning goes "physicists can't explain wave function collapse; neuroscientists can't explain consciousness; therefore ..." and then some nonsense.

A less magical way to think about quantum measurement is in terms of *interaction* between the quantum state of a particle and a classical measurement apparatus which can be in one of n possible states. Because of the relative size of the two interacting systems, the state $|\psi\rangle$ is forced to "align" with one of the n possible states of the measurement apparatus. A quantum measurement is an interaction that creates classical information and destroys quantum information. By measuring, we obtained the measurement outcome i, but disturbed the initial state $|\psi\rangle$, forcing it into the "aligned with Π_i"-state $|\psi_i'\rangle$. We can still carry out more experiments with the post-measurement state $|\psi_i'\rangle$, but it's not the same as the initial state $|\psi\rangle$. Specifically, we've lost all the information about $|\psi\rangle$ that used to lie in the subspace $(\mathbb{1} - \Pi_i)$.

Example 4 In Figure 10.13 a state vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is being measured with photo detectors modelled as projectors given by

$$\begin{array}{ll} \Pi_0 = |0\rangle\langle 0| & \sum_j \Pi_j = \mathbb{1} \\ \Pi_1 = |1\rangle\langle 1| & \end{array}$$

$$\begin{aligned} \Pr(\{0\}|\psi) &\equiv \Pr(\{\text{outcome 0 given state } |\psi\rangle\}) \\ &\equiv \langle\psi| \quad \Pi_0 \quad |\psi\rangle \\ &= \langle\psi| \quad |0\rangle\langle 0| \quad |\psi\rangle \\ &= (\bar{\alpha}\langle 0| + \bar{\beta}\langle 1|)\,|0\rangle\langle 0|\,(\alpha|0\rangle + \beta|1\rangle) \\ &= \bar{\alpha}\alpha \\ &= |\alpha|^2. \end{aligned}$$

The probability of outcome 1 is $\Pr(\{1\}|\psi) \equiv \langle\psi|\Pi_1|\psi\rangle = |\beta|^2$. The state of the quantum system after the measurement is in one of two possible states: $|\psi_0'\rangle = |0\rangle$ or $|\psi_1'\rangle = |1\rangle$.

Example 5 Consider the measurement $\{\Pi_+, \Pi_-\}$ that consists of the projectors onto the Hadamard basis:

$$\begin{array}{ll} \Pi_+ = |+\rangle\langle +| & \Pi_+ + \Pi_- = \mathbb{1} \\ \Pi_- = |-\rangle\langle -| & \end{array}$$

Given the quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probability of outcome "+" is given by

$$\begin{aligned}
\Pr(\{+\}|\psi) &\equiv \|\Pi_+|\psi\rangle\|^2 \\
&= \||+\rangle\langle+||\psi\rangle\|^2 \\
&= \||+\rangle\langle+|(\alpha|0\rangle + \beta|1\rangle)\|^2 \\
&= \||+\rangle(\alpha\langle+|0\rangle + \beta\langle+|1\rangle)\|^2 \\
&= \left\||+\rangle\left(\alpha\tfrac{1}{\sqrt{2}} \;\; + \beta\tfrac{1}{\sqrt{2}}\right)\right\|^2 \\
&= \frac{(\alpha+\beta)^2}{2}\||+\rangle\|^2 \\
&= \frac{(\alpha+\beta)^2}{2}.
\end{aligned}$$

The probability of outcome "−" is $\Pr(\{-\}|\psi) = \frac{(\alpha-\beta)^2}{2}$. The state of the quantum system after the measurement is in one of two possible states: $|\psi'_+\rangle = |+\rangle$ or $|\psi'_-\rangle = |-\rangle$.

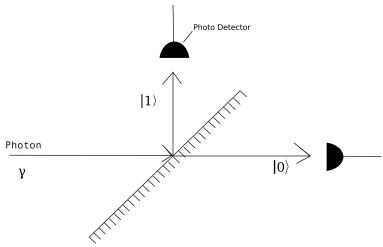


Figure 10.13: The state of a photon after encountering a $(1-\alpha)$-silvered mirror is $|\gamma\rangle = \alpha|0\rangle + \beta|1\rangle$. The probability that the horizontal photodetector "clicks" is $|\alpha|^2$, and is obtained by projecting $|\gamma\rangle$ on the subspace $|0\rangle\langle 0|$. The probability that the top photodetector clicks is equal to $|\beta|^2$, and is obtained by projecting $|\gamma\rangle$ on the subspace $|1\rangle\langle 1|$.

The measurement process is a fundamental aspect of the quantum models. You have to get used to the idea that measurements change systems' states. That's not magic, but simply due to the relative size of the systems (tiny quantum particles and huge measurement apparatus) and the fact we're forcing them to interact.

Composite quantum systems

So far we discussed state preparation, quantum operations, and quantum measurements of individual qubits. It's now time to discuss quantum models for systems made up of multiple qubits.

Classically, if we have two bits $b_1 \in \{0,1\}$ and $b_2 \in \{0,1\}$, we can concatenate them to obtain a bit string $b_1 b_2 \in \{0,1\}^2$, which can have one of four possible values: 00, 01, 10, and 11. The combined state of two qubits $|\varphi_1\rangle \in \mathbb{C}^2$ and $|\varphi_2\rangle \in \mathbb{C}^2$ is the *tensor product state* $|\varphi_1\rangle \otimes |\varphi_2\rangle$ in the four-dimensional *tensor product space* $\mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^4$. A basis for the tensor product space can be obtained by taking all possible combinations of the basis elements for the individual qubits: $\{|0\rangle \otimes |0\rangle,\ |0\rangle \otimes |1\rangle,\ |1\rangle \otimes |0\rangle,\ |1\rangle \otimes |1\rangle\}$.

Postulate 4. The state space of a composite quantum system is the tensor product of the state spaces of the individual systems. If you have systems $1, 2, \ldots, n$ in states $|\varphi_1\rangle, |\varphi_2\rangle, \cdots, |\varphi_n\rangle$, then the state of the composite system is $|\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \cdots \otimes |\varphi_n\rangle$.

Postulate 4 tells us how the state spaces of different quantum systems may be combined to give a description of the composite system. A lot of the interesting quantum applications involve operations on multiple qubits and are described by vectors in a tensor product space, so we better look into this "$\otimes$"-thing more closely.

Tensor product space You may not have heard of the *tensor product* before, but don't worry about it. The only scary part is the symbol "$\otimes$." A *tensor product space* consists of all possible combinations of the basis vectors for the two subspaces. For example, consider two qubits $|\varphi_1\rangle \in V_1 \equiv \mathbb{C}^2$ and $|\varphi_2\rangle \in V_2 \equiv \mathbb{C}^2$. We'll denote the standard basis for V_1 as $B_1 \equiv \{|0\rangle_1, |1\rangle_1\}$ and the standard basis for V_2 as $B_2 \equiv \{|0\rangle_2, |1\rangle_2\}$. The tensor product space $B_{12} \equiv V_1 \otimes V_2$ is four-dimensional and this is a basis for it:

$$B_{12} \equiv \big\{|0\rangle_1 \otimes |0\rangle_2,\ |0\rangle_1 \otimes |1\rangle_2,\ |1\rangle_1 \otimes |0\rangle_2,\ |1\rangle_1 \otimes |1\rangle_2\big\}.$$

This level of subscripts and the explicit use of the symbol $\otimes$ really hurts the eyes (and the hand if you have to use this notation when solving problems). It is therefore customary to drop the subscripts indicating which vector space the vector comes form, omitting the tensor product symbol, and drawing a single ket which contains a "string" of indices

$$|a\rangle_1 \otimes |b\rangle_2 = |a\rangle \otimes |b\rangle = |a\rangle|b\rangle = |ab\rangle.$$

The basis for the tensor product space $B_{12} \equiv V_1 \otimes V_2$ looks much nicer using the simplified notation:

$$B_{12} \equiv \big\{|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle\big\}.$$

Tensor product of two vectors Suppose we're given two qubits whose states are described by the following vectors:

$$|\varphi_a\rangle_1 = \alpha_1|0\rangle_1 + \beta_1|1\rangle_1, \qquad |\varphi_b\rangle_2 = \alpha_2|0\rangle_2 + \beta_2|1\rangle_2,$$

Note the subscripts indicate which system we're describing: $|0\rangle_1$ is the state $|0\rangle$ for the first qubit, while $|0\rangle_2$ is the state $|0\rangle$ of the second qubit.

The state of the combined system is tensor product state $|\varphi_{ab}\rangle_{12} = |\varphi_a\rangle_1 \otimes |\varphi_b\rangle_2$, which is computed by combining all possible combinations of the coefficients of $|\varphi_a\rangle_1$ and the coefficients of $|\varphi_b\rangle_2$:

$$(\alpha_1,\ \beta_1)_{B_1} \otimes (\alpha_2,\ \beta_2)_{B_2} = (\alpha_1\alpha_2,\ \alpha_1\beta_2,\ \beta_1\alpha_2,\ \beta_1\beta_2)_{B_{12}}\,.$$

The notion of "all possible linear combinations" is easier to see by considering the tensor product operation in terms of the basis vectors:

$$\begin{aligned}
|\varphi_{ab}\rangle_{12} &= |\varphi_a\rangle_1 \otimes |\varphi_b\rangle_2 \\
&= (\alpha_1|0\rangle_1 + \beta_1|1\rangle_1) \otimes (\alpha_2|0\rangle_2 + \beta_2|1\rangle_2) \\
&= \alpha_1\alpha_2|0\rangle_1|0\rangle_2 + \alpha_1\beta_2|0\rangle_1|1\rangle_2 + \beta_1\alpha_2|1\rangle_1|0\rangle_2 + \beta_1\beta_2|1\rangle_1|1\rangle_2 \\
&= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle \\
&= (\alpha_1\alpha_2,\ \alpha_1\beta_2,\ \beta_1\alpha_2,\ \beta_1\beta_2)_{B_{12}}\,.
\end{aligned}$$

Where $B_{12} \equiv \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ is the standard basis for the tensor product space.

Dimension counting A quantum state that consists of n qubits can represent any unit length vector in $\mathbb{C}^{2n}$. That's an insanely big state space—a huge $2n$-dimensional playground. In comparison, a classical bitstring of length n can take on one of 2^n values.

A very large state space does not necessarily make a model more successful, but the large dimension of the tensor product space suggest many new possibilities. Much of the recent excitement in the area of quantum computing is based on the promise of using the qubits of a quantum computer to perform computations in very large quantum state spaces. Think how much more powerful quantum states are compared to binary strings. Adding an extra bit to a classical registers doubles the number of states it can represent. Adding a qubit to a quantum register adds a whole two-dimensional subspace to the quantum state space.

We shouldn't get carried away with enthusiasm, because with great state space comes great noise! It is easy to imagine n qubits in a row, but building a physical system which can store n qubits and protect them from noise is a much more difficult task. Another bottleneck in quantum computing is the difficulty of extracting information from

quantum systems. The quantum state space of n qubits is $\mathbb{C}^{2n}$, but projective measurements of the form $\{\Pi_1, \Pi_2, \ldots, \Pi_m\}$ can only obtain the one answer to a question with m possible classical outcomes ($m \leq 2^n$). The information we learn about a quantum state through measurement is directly proportional to the disturbance we cause to the state. We'll learn more about theoretical and practical considerations for quantum computing in Section 10.8.

Exercises

E10.6 Show that the quantum state $|0\rangle|1\rangle - |1\rangle|0\rangle$ is equal to the quantum state $|+\rangle|-\rangle - |-\rangle|+\rangle$.

Quantum entanglement

At the risk of veering further off-topic for a linear algebra book, we'll now say a few words about quantum states which are *entangled*. In particular we'll discuss the properties of the *entangled state* $|\Psi_-\rangle \equiv \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Some of the most fascinating results in quantum information science make use of pre-shared entangled states. Entanglement is some really crazy stuff, and I mean really crazy, not just science-journalism crazy.

In Section 8.9 we discussed how shared secret key between two parties, Alice and Bob, is a *communication resource* that can be used to achieve private communication (using the one-time pad cryptosystem). You can think of entanglement as another type of communication resource: a stronger-than-classical correlation between two parts of a quantum system. One half of system is held by Alice, the other half is held by Bob. When the collective state of a two-qubit quantum systems is in the entangled state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$, measurements on the individual qubits will produce anti-correlated results *in any basis*.

Example 7 The Einstein-Podolsky-Rosen (EPR) state is a two-qubit quantum state with interesting *nonlocal* properties. We assume that Alice holds one half of the quantum state, Bob holds the other half of the following state:

$$|\Psi_-\rangle^{AB} \equiv \tfrac{1}{\sqrt{2}}|0\rangle^A|1\rangle^B - \tfrac{1}{\sqrt{2}}|1\rangle^A|0\rangle^B.$$

Note the use of superscript to denote which party holds that part of the system.

Let's analyze the results of different measurement Alice and Bob can perform on this state. If Alice chooses to measures her system in the basis $\{|0\rangle, |1\rangle\}$, the projection operators that correspond to the

two outcomes are

$$\Pi_0^{AB} = |0\rangle\langle 0|^A \otimes \mathbb{1}^B \qquad \text{and} \qquad \Pi_1^{AB} = |1\rangle\langle 1|^A \otimes \mathbb{1}^B.$$

Since we're measuring only Alice's half of the state, the measurement acts like the identity operator on Bob's half of the state. There's a 50-50 chance of outcomes 0 and 1, and depending on the outcome, the post-measurement state of the system will be either $|0\rangle^A|1\rangle^B$ or $|1\rangle^A|0\rangle^B$. If Bob then measures his half of the system, he'll be sure to obtain the opposite outcome of Alice's. In other words, the measurement outcomes that Alice and Bob obtain are perfectly anti-correlated.

What if Alice and Bob choose to measure their respective halves of the EPR state $|\Psi_-\rangle^{AB}$ in the basis $\{|+\rangle, |-\rangle\}$? Using some basic calculations (see Exercise **E10.6**), we can express the EPR state $|\Psi_-\rangle^{AB}$ in terms of the basis $\{|+\rangle, |-\rangle\}$ as follows:

$$\tfrac{1}{\sqrt{2}}|0\rangle^A|1\rangle^B - \tfrac{1}{\sqrt{2}}|1\rangle^A|0\rangle^B = |\Psi_-\rangle^{AB} = \tfrac{1}{\sqrt{2}}|+\rangle^A|-\rangle^B - \tfrac{1}{\sqrt{2}}|-\rangle^A|+\rangle^B.$$

Observe the state $|\Psi_-\rangle^{AB}$ has the same structure in the Hadamard basis as in the standard basis. Thus, Alice and Bob's measurement outcomes will also be perfectly anti-correlated when measuring in the Hadamard basis.

A state $|\psi\rangle^{AB}$ is called *entangled* if it cannot be written as a tensor product of quantum states $|\phi\rangle^A \otimes |\varphi\rangle^B$, where $|\phi\rangle^A$ describes the state held by Alice and $|\varphi\rangle^B$ the state help by Bob. The EPR state $|\Psi_-\rangle^{AB}$ is *entangled*, which means it cannot be written as a tensor product of the quantum states of individual qubits $|\phi\rangle^A \otimes |\varphi\rangle^B$:

$$|0\rangle\otimes|1\rangle - |1\rangle\otimes|0\rangle \;\neq\; (a|0\rangle + b|1\rangle)\otimes(c|0\rangle + d|1\rangle),$$

for any $a, b, c, d \in \mathbb{C}$. Since we cannot describe the EPR state $|\Psi_-\rangle^{AB}$ as the tensor product of two local states $|\phi\rangle^A$ and $|\varphi\rangle^B$, we say it requires a *nonlocal* description, which is another way of saying $|\Psi_-\rangle^{AB}$ is entangled.

There is something strange about the EPR state. If Alice measures her half of the state and finds $|0\rangle$, then we know immediately that Bob's state will be $|1\rangle$. The collapse in the superposition on Alice's side immediately causes a collapse of the superposition on Bob's side. Note that Bob's collapse will occur *immediately*, no matter how far Bob's system is from Alice's. This is what the authors Einstein, Podolsky, and Rosen called "spooky action at a distance." How could Bob's system "know" to always produce the opposite outcome even at the other end of the universe?

Now imagine you have a whole bunch of physical systems prepared in the EPR state. Alice has the left halves of the EPR pairs, Bob

has all the right halves. This is a communication resource we call *shared entanglement*. Many of the quantum information protocols make use of shared entanglement between Alice and Bob, to achieve novel communication tasks.

So far we discussed entangled states from their measurement outcomes as answers to questions. From the point of view of information, a system is entangled whenever you know more about the system as a whole than about its parts. But who has seen this entanglement? Does $|\Psi_-\rangle$ really exist? How can we know that $|\Psi_-\rangle$ isn't just some math scribble on a piece of paper? To quell your concerns, we'll give a real-world physics example where we knowing more about the whole system than about its constituent parts.

Physics example We'll now describe a physical process that leads to the creation of an entangled quantum state. Consider a quantum particle p that *decays* into two quantum subparticles p_a and p_b. The decay process obeys various physics conservation laws, in particular, the total spin angular momentum before and after the decay must be conserved. Suppose the particle p had zero spin angular momentum before the decay, then the conservation of angular momentum principle dictates the subparticles p_a and p_b must have opposite spin. The spin angular momentum of a each particle is in an unknown direction (up, down, left, right, in, or out), but whatever spin direction we measure for p_a, the spin angular momentum of p_b will immediately take on the opposite direction.

The general scenario discussed above describes what happens if a Helium atom were to explode, and the two electrons in it's ground state fly off to distant sides of the universe. The two electrons will have opposite spins, but we don't know the directions of the individual spins. The only thing we know is their total spin is zero, since the ground state of the Helium atom had spin zero. This is how we get the "anti-correlation in *any* basis" aspect of quantum entanglement.

Summary

We can summarize the new concepts of quantum mechanics and relate them to the standard concepts of linear algebra in the following table:

$$\begin{aligned} \text{quantum state} &\Leftrightarrow \text{vector } |v\rangle \in \mathbb{C}^d \\ \text{evolution} &\Leftrightarrow \text{unitary operations} \\ \text{measurement} &\Leftrightarrow \text{projections} \\ \text{composite system} &\Leftrightarrow \text{tensor product} \end{aligned}$$

The quantum formalism embodied in the four postulates discussed above has been applied to describe many physical phenomena. Using complex vectors to represent quantum states leads to useful models and predictions for experimental outcomes. In the next section we'll use the quantum formalism to analyze the outcomes of the polarization lenses experiment.

In addition to the applications of quantum principles, studying the structure in quantum mechanics states and operations is an interesting field on its own. An example of fundamentally new quantum idea is existence of *entangled* quantum states, which are states of a composite quantum system $|\Phi\rangle_{12} \in V_1 \otimes V_2$ that cannot be written as a tensor product of local quantum states of the individual systems: $|\Psi\rangle_{12} \neq |\phi\rangle_1 \otimes |\varphi\rangle_2$. Later on in this section discuss an interesting application of quantum entanglement, as part of the *quantum teleportation* protocol, illustrated in Figure 10.23 (page 468).

Exercises

E10.7 Compute probability of outcome "$-$" for the measurement $\{\Pi_+, \Pi_-\} = \{|+\rangle\langle+|, |-\rangle\langle-|\}$ performed on the quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Links

[Compact set of notes on QM written by a physicist]
`http://graybits.biz/notes/quantum_mechanics/preface`

[Lecture series on QM written by a computer scientist]
`http://scottaaronson.com/democritus/lec9.html`

10.6 Polarizing lenses experiment revisited

Let's revisit the polarizing lenses experiment, this time modelling the polarization states of photons as two-dimensional complex vectors. We can define the state of a horizontally polarized photon as $|H\rangle \equiv (1,0)^\mathsf{T}$ and the vertical polarization state as $|V\rangle \equiv (0,1)^\mathsf{T}$. This choice corresponds to the observation that horizontal and vertical polarizations are complementary, which we model as orthogonal state vectors. Starting with unpolarized light, and passing it through a polarizing lens allows us to prepare photons in a chosen state:

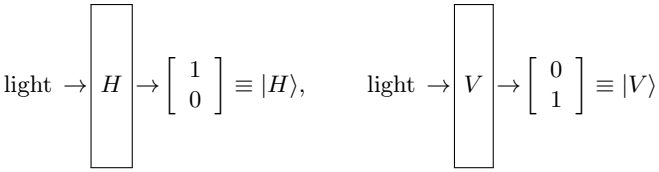


Figure 10.14: State preparation procedure for photons with horizontal or vertical polarization.

Placing a polarizing lens in the path of a photon corresponds to the measurement $\{\Pi_{\leftarrow}, \Pi_{\rightarrow}\}$ of the photon's polarization state. The two possible outcomes of the measurements are: "photon goes through" and "photon is reflected." Each photon that hits a polarizing lens is asked the question "are you horizontally or vertically polarized?" and forced to decide.

Figure 10.15 shows the projection operators that correspond to a measurement using a horizontally polarizing lens. The outcome "goes through" corresponds to the projection operator $\Pi_{\rightarrow} = |H\rangle\langle H| \equiv \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. The outcome "is reflected" corresponds to the projection matrix $\Pi_{\leftarrow} = |V\rangle\langle V| \equiv \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

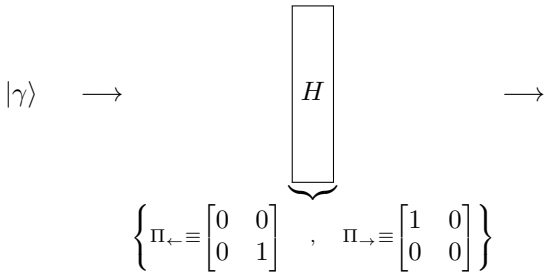


Figure 10.15: A horizontally polarizing lens corresponds to the quantum measurement $\{\Pi_{\leftarrow}, \Pi_{\rightarrow}\}$. An incoming photon in the state $|\gamma\rangle$ is asked to choose one of the two alternative paths. With probability $\|\Pi_{\rightarrow}|\gamma\rangle\|^2$, the photon will go through the H-polarizing lens and become horizontally polarized $|\gamma'\rangle = (1,0)^{\mathsf{T}}$. With probability $\|\Pi_{\leftarrow}|\gamma\rangle\|^2$, it will be reflected.

Recall that the probability of outcome i in a quantum measurement $\{\Pi_i\}$ is given by the expression $\|\Pi_i|\gamma\rangle\|^2$, where $|\gamma\rangle$ is the state of the incoming photon. Knowing the projection operators that correspond to the "goes through" and "is reflected" outcomes allows us to predict the probability that photons with a given state will make it though the lens. The setup shown in Figure 10.16 illustrates a simple two-step experiment in which states prepared in the horizontally polarized state $|H\rangle = (1,0)^{\mathsf{T}}$ arrive at a V-polarizing lens. The probability of

photons making it through the V-polarizing lens is

$$\begin{aligned}\Pr(\{\text{pass through } V \text{ lens given state } |H\rangle\}) &= \| |V\rangle\langle V||H\rangle \|^2 \\ &= \left\| \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|^2 = 0.\end{aligned}$$

Indeed, this is what we observe in the lab—all $|H\rangle$ photons are are reject by the V-polarizing lens.

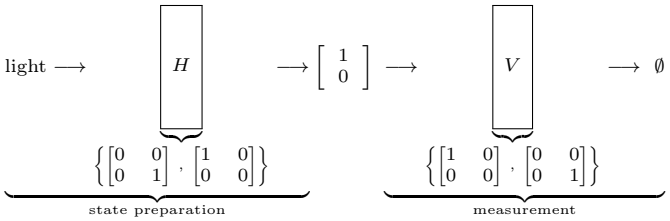


Figure 10.16: Photons prepared in the state $|H\rangle = (1,0)^\mathsf{T}$ are rejected by the V-polarizing lens because their states have zero overlap with the projector $\Pi_\rightarrow = |V\rangle\langle V|$.

Let's now use the quantum formalism to analyze the results of the three-lenses experiment which we saw earlier in the chapter. Figure 10.17 shows the optical circuit which consists of a state preparation step and two measurement steps. The diagonally polarizing lens placed in the middle of the circuit only allows through photons that have 45°-diagonal polarization: $|D\rangle \equiv \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\mathsf{T}$. The projection operator associated with the "goes through" outcome of the diagonal polarization measurement is

$$\Pi_\rightarrow = |D\rangle\langle D| = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The post-measurement state of photons that make it through the diagonally polarizing lens will be $|D\rangle \equiv \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\mathsf{T}$.

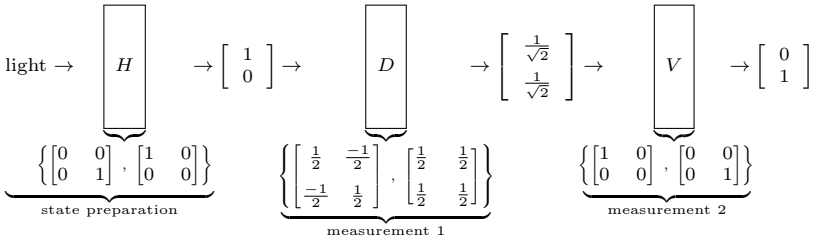


Figure 10.17: Photons prepared in the state $|H\rangle$ are subjected to two sequential measurements: a diagonal polarizing measurement D followed by a vertical polarization measurement V. The projection operators for "is reflected" and "goes though" outcomes are indicated in each step.

The photons that made it through the middle lens are in the state $|D\rangle$. Let's analyze what probability of photons making it through the V-polarizing lens:

$$\begin{aligned}
\Pr(\{\text{pass through } V \text{ lens given state } |D\rangle\}) &= \| |V\rangle\langle V| |D\rangle \|^2 \\
&= \left\| \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\|^2 \\
&= \left\| \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\|^2 \\
&= 0^2 + \left(\tfrac{1}{\sqrt{2}}\right)^2 \\
&= \frac{1}{2}.
\end{aligned}$$

The overall probability of a photon making it through until the end is $\frac{1}{4}$.

This probability is consistent with the light intensity observations we see in Figure 10.8 (page 424). Observe the difference in interpretation: we previously referred to the optical power $P = 0.5$ after the first measurement, and optical power $P = 0.25$ after the second measurement, whereas the quantum formalism refers to probabilities $p = 0.5$ and $p = 0.25$ for the photons to reach different parts of the circuit.

Discussion

The three-lenses experiment serves to illustrates some key aspects of the quantum formalism. When performed with a beam of light, the outcomes of the experiment can be explained using the classical theory of electromagnetic waves. The light beam consists of so many

photons that it behaves like continuous quantity that can be split: part of the wave goes through the lens, and another part gets reflected. The projections to the "goes through" orientation performed by the polarizing lenses can easily be understood if we model the light beam as a classical wave. Therefore, the table-top experiment we performed cannot be seen as a proof that quantum mechanics description of reality is necessary.

The same experiment can be reproduced with *single photon sources*, which behave like really really weak laser pointers, emitting only one photon at a time. The classical explanation runs into trouble explaining what happens, since a single photon cannot be subdivided into parts. Rather than seeing this as a "failure" of classical wave theory, we can see it as a resounding success, since understanding waves allows you to understand quantum theory better.

The polarizing lenses experiment is inspired by the famous *Stern–Gerlach experiment*. The demonstration and reasoning about the observed outcomes is very similar, but the experiment is performed with the magnetic spin of silver atoms. I encourage you to learn more about the original Stern–Gerlach experiment.

`https://en.wikipedia.org/wiki/Stern-Gerlach_experiment`
`https://youtube.com/watch?v=rg4Fnag4V-E`

10.7 Quantum physics is not that weird

Without a doubt, you have previously heard that quantum mechanics is mysterious, counterintuitive, and generally "magical." It's not *that* magical, unless, of course, vector operations count as magic. In this section we'll single out three so called "weird" aspects of quantum mechanics: superposition, interference, and the fact that quantum measurements affect the states of systems being measured. We'll see these aspect of quantum mechanics are actually not that weird. This is a necessary "demystification" step so we can get all the quantum sensationalism out of the way.

Quantum superposition

Classical binary variables (bits) can have one of two possible values: 0 or 1. Examples of physical systems that behave like bits are electric switches that can be either open or closed, digital transistors that either conduct or don't conduct electricity, and capacitors that are either charged or discharged.

A quantum bit (qubit), can be both 0 and 1 *at the same time*. Wow! Said this way, this surely sounds impressive and mystical, no?

But if we use the term *linear combination* instead of "at the same time," the quantum reality won't seem so foreign. A quantum state is a linear combination of the basis states. This isn't so crazy, right? The *superposition principle* is a general notion in physics that is not specific to quantum phenomena, but applies to all systems described by differential equations. Indeed, superpositions exist in many classical physics problems too.

Example Consider a mass attached to a spring that undergoes simple harmonic motion. The differential equation that governs the motion of the mass is $x''(t)+\omega^2 x(t) = 0$. This equation has two solutions: $x_0(t) = \sin(\omega t)$ and $x_1(t) = \cos(\omega t)$, corresponding to two different starting points of the oscillation. Since both $x_0(t)$ and $x_1(t)$ satisfy the equation $x''(t) + \omega^2 x(t) = 0$, any linear combination of $x_0(t)$ and $x_1(t)$ is also a solution. Thus, the most general solution to the differential equation is of the form:

$$x(t) = \alpha x_0(t) + \beta x_1(t) = \alpha \sin(\omega t) + \beta \cos(\omega t).$$

Usually we combine the sin and cos terms and describe the equation of motion for the mass-spring system in the equivalent form $x(t) = A\cos(\omega t + \phi)$, where A and ϕ are computed from α and β. In science-journalist speak however, the mass-spring system would be described as undergoing both sin motion and cos motion at the same time! Do you see how ridiculous this sounds?

The notion of *quantum superposition* is simply a consequence of the general superposition principle for differential equations. If the quantum states $|0\rangle$ and $|1\rangle$ both represent valid solutions to a quantum differential equation, then the state of the system can be described as a linear combination of these two solutions:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

The observation that "$|\psi\rangle$ is both $|0\rangle$ and $|1\rangle$ at the same time" is not wrong, it's just not very useful. It's much more precise to describe the quantum state $|\psi\rangle$ as a linear combination.

Interference

Unlike particles that bounce off each other, waves can co-exist in the same place. The resulting wave pattern is the sum of the constituent waves. Quantum particles behave similarly to waves in certain experiments, and this can lead to *interference* between quantum systems.

The prototypal example of interference is Young's double-split experiment, in which particles passing through two thin slits interact

with each other causing an interference pattern of alternating bright and dark spots on a screen. Classical physics models assume particles behave like tiny point-like balls that bounce off each other whenever they come in contact. The prediction of a classical model is that particles will appear on the screen in two bright peaks, directly facing the two slits.

In contrast, the quantum model of a particle describes it as a travelling energy pulse that exhibits wave-like properties.[2] In a quantum model, the particles coming through the slits behave like waves and can combine constructively or destructively, depending on the relative distances travelled by the particles. Similar interference patterns occur whenever waves combine, as in the example of waves on the surface of a liquid, or sound waves.

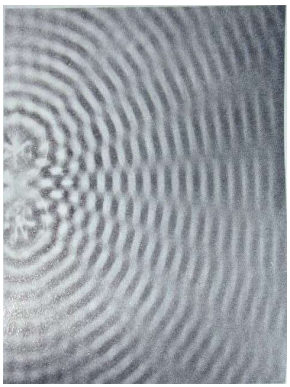


Figure 10.18: The waves emitted by two synchronized sources form an interference pattern. Observe the stripes of destructive interference where the waves meat "out of sync" (peak to troughs) and cancel each other out.

Performing the experiment reveals a pattern of bright and dark stripes (called fringes) on the screen in support of the quantum model. The locations of the dark fringes corresponds exactly to the places where particles coming from the two slits arrive "out of sync," and combine destructively:

$$|\psi\rangle - |\psi\rangle = 0.$$

This case corresponds to the dark fringes on the screen, where no particles arrive.

The idea that one wave can cancel another wave is not new. What *is* new is the observation that particles behave like waves and can interfere with each other. That's definitely new. Interference was one of

[2]This is where the name *wave function* comes from.

the first puzzling effect of quantum systems that was observed. Observations from interference experiments forced physicists to attribute wave-like properties even to particles.

[Video demonstration of Young's double-split experiment]
`https://youtube.com/watch?v=qCmtegdqOOA`

Measurement of a system affects the system's state

Another "weird" aspect of quantum mechanics is the notion that quantum measurements can affect the states of the systems being measured. This is due to the energy scale and the size of systems where quantum physics comes into play, and not due to some sort of "quantum magic." Let's see why.

When we think about physical systems on the scale of individual atoms, we can't continue to see ourselves (and our physical measurement apparatuses) as a passive observers of the systems. We need to take into account the *interactions* between quantum systems and the measurement apparatus. For example, in order to see a particle, we must bounce off at least one photon from the particle and then collect the rebounding photon at a detector. Because the momentum and the energy of a single particle are tiny, the collision with the photon as part of the experiment will make the particle recoil. The particle recoils from the collision with the observing photon because of the conservation of momentum principle. Thus, the measurement might send the particle flying off in a completely different direction. This goes to show that measurements affecting the sate of systems being measured is not some magical process but simply due to tiny energies of the particles being observed.

Wave functions

The quantum mechanics techniques we discussed in this chapter are good for modelling physical systems that have discrete sets of states. In *matrix quantum mechanics*, quantum states are described by a vectors in finite-dimensional complex inner product spaces. Other physics problems require the use of *wave function quantum mechanics* in which quantum states are represented as complex-valued functions of space coordinates $\vec{r} = (x, y, z)$. Instead of the dot product between vectors, the inner product for wave functions is $\langle f(\vec{r}), g(\vec{r}) \rangle = \iiint_{\mathbb{R}^3} \overline{f(\vec{r})} g(\vec{r}) \, d^3\vec{r}$. This may seem like a totally new ball game, but actually calculations using wave functions are not too different from inner product calculations we used to compute Fourier transformations in Section 8.11.

It's beyond the scope of the current presentation to discuss *wave functions* in detail, but I want to show you an example of a calculation with wave functions, so you won't say that I didn't show you some proper physics stuff. The ground state of the Hydrogen atom is described by the wave function $\psi(\vec{r}) = \frac{1}{\sqrt{\pi a^3}} \exp(-r/a)$, where $r = \|\vec{r}\|$. The probability of finding the electron at position $\vec{r}$ from the proton is described by the inner product $\overline{\psi(\vec{r})}\psi(\vec{r}) \equiv |\psi(\vec{r})|^2$:

$$\Pr(\{\text{finding electron at } \vec{r}\,\}) = |\psi(\vec{r})|^2.$$

Since $\psi(\vec{r})$ depends only on the distance r, we know the wave function has a spherically symmetric shape, as illustrated in Figure 10.19.

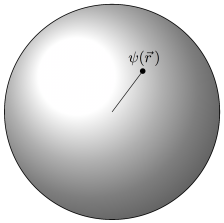


Figure 10.19: The s orbital of an electron is spherically symmetric.

We'll now check if $|\psi(\vec{r})|^2$ is a properly normalized probability density function. Integrating the probability density function $|\psi(\vec{r})|^2$ over all of $\mathbb{R}^3$ should give one. We'll use spherical coordinates (r, ϕ, θ) to solve this problem instead of cartesian coordinates (x, y, z). In spherical coordinates, the volume of a thin slice from the surface of a sphere of width $d\theta$, height $d\phi$, and thickness dr is given by $r^2 \sin\phi \, d\phi \, d\theta \, dr$. If you haven't seen spherical coordinates before, don't worry about this expression too much. The conversion factor $r^2 \sin\phi$ is just some trickery needed to convert the "small piece of volume" $d^3\vec{r} = dx \, dy \, dz$ to an equivalent small piece of volume in spherical coordinates $d^3\vec{r} = r^2 \sin\phi \, d\phi \, d\theta \, dr$. See **P4.20** for the derivation.

We'll split the triple integral into two parts: an integral that depends only on the radius r, and double integral over all angles ϕ and θ:

$$\begin{aligned} p_{\text{total}} &= \iiint_{\mathbb{R}^3} |\psi(\vec{r})|^2 \, d^3\vec{r} \\ &= \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} |\psi(r)|^2 \, r^2 \sin\phi \, d\phi \, d\theta \, dr \\ &= \left(\int_0^{\infty} \frac{1}{\pi a^3} \exp(-2r/a) \, r^2 \, dr \right) \left(\int_0^{2\pi} \int_0^{\pi} \sin\phi \, d\phi \, d\theta \right) \end{aligned}$$

$$= \left(\int_0^\infty \frac{1}{\pi a^3} \exp(-2r/a)\ r^2\, dr \right) (4\pi)$$
$$= \int_0^\infty \underbrace{\frac{4}{a^3} \exp\left(\frac{2r}{a}\right) r^2}_{p(r)}\ dr.$$

The expression $p(r) = \frac{4}{a^3} \exp(-2r/a) r^2$ describes the probability of finding the electron at a distance r from the centre of the nucleus. We can complete the calculation of total probability by taking the integral of $p(r)$ from $r = 0$ to $r = \infty$:

$$\begin{aligned} p_{\text{total}} &= \int_0^\infty p(r)\, dr \\ &= \frac{4}{a^3} \int_0^\infty \exp\left(\frac{2r}{a}\right) r^2\, dr \\ &= 1. \end{aligned}$$

The purpose of working through this wave function calculation is to give you an idea of the complex calculations physicists have to do on a day to day basis using the wave function formalism. In comparison, the matrix formalism for quantum mechanics is much simpler, involving only basic linear algebra calculations.

10.8 Quantum mechanics applications

What can we do using quantum physics that we couldn't do using classical physics? What can we compute with qubits that we couldn't compute with bits? After learning the quantum formalism, it's time to see if it's useful for something or not. Below we present some areas of physics and computer science that wouldn't exist without the laws of quantum mechanics.

Particle physics

The quantum mechanics formalism we discussed above is not well suited for describing the behaviour of high energy particles. The best current model for to describing high energy physics is called *quantum field theory*, and models each fundamental particle as a disturbance in a *particle field*. Particles (disturbances) and antiparticles (negatives of disturbances) can be created and destroyed, and interact with other particle fields. It's a bit like chemistry where different combinations of atoms get transformed into other combinations of atoms, but instead of atoms we have elementary particles like quarks and leptons. The same way *Mendeleev's periodic table* gives us a catalogue of all the

available atoms, the *Standard model* of particle physics gives us the catalogue of elementary particles, which combine to form particles like protons and neutrons and transform into other combinations of particles in high-energy physics experiments.

As the name suggests, high energy physics is one extreme of an energy continuum. At the low energy scale of the continuum the rules of chemistry rule. Chemical reactions describe how molecules transform into other molecules, which basically represent different ways electrons can be shared between a bunch of atoms. At higher energies, atoms get "stripped" of their electrons because they have so much energy that they're not bound to the nucleus anymore. At this point the laws of chemistry stop being relevant, since electrons are freely flying around and the molecules are no longer together. Exit chemistry; enter nuclear physics. Nuclear physics studies the different combinations of protons and neutrons that can form the nuclei of different atoms. A nuclear reaction is like a chemical reaction, but instead of chemical molecules the reactant and products are different types of nuclei. An example of nuclear reaction is the fusion of two heavy hydrogen nuclei to form a helium nucleus. At higher energy still, even protons and neutrons can break apart, and the analysis shifts to interactions between elementary particles like leptons, bosons, neutrinos, quarks, and photons. This is the domain of high energy physics.

The basic postulates of quantum mechanics still apply in quantum field theory, but the models become more complicated since we assume even the interactions between particles are quantized. You can think of the basic quantum mechanics described in this chapter as learning the alphabet, and quantum field theory as studying Shakespeare, including the invention of new words. Studying quantum field theory requires new math tools like path integrals, new intuitions like symmetry observations, and new computational tricks like *renormalization*. The essential way of thinking about photons, electrons, and the interactions between them can obtained by reading Richard Feynman's short book titled *QED*, which stands for *quantum electrodynamics*. In this tiny book, Feynman uses an analogy of a "tiny clock" attached to each particle to explain the phase $e^{i\theta}$ of its wave function. Starting from this simple analogy the author works his way up to explaining concepts from graduate-level quantum field theory like path integrals. I highly recommended this book. It's your chance to learn from one of the great scientists in the field and one of the best physics teachers of all times.

[The Standard Model of particle physics]
`https://en.wikipedia.org/wiki/Standard_Model`

[Nuclear fusion is the way energy is generated inside stars]
`https://en.wikipedia.org/wiki/Nuclear_fusion`

[BOOK] Richard P. Feynman. *QED: The strange theory of light and matter*. Princeton University Press, 2006, ISBN 0691125759.

Solid state physics

Physicists have been on a quest to understand the inner structure of materials ever since the first days of physics. Some examples of applications developed based on this understanding include semiconductors, lasers, photovoltaic batteries (solar panels), light emitting diodes (LEDs), and many other. These applications all depend on specially engineered conductivity properties of materials. Indeed, thinking about the conductivity of materials can give us a very good picture of their other properties. We can classify materials into the following general groups: insulators, metals, and semi-conductors. These three categories correspond to materials with different *energy band structure*.

Insulators are the most boring type of material. A good example of an insulator is glass—just an amorphous clump of silica (SiO_2). The term *glass* is used more generally in physics to describe any material whose molecules are randomly oriented and don't have any specific crystal structure. Conductors are more interesting. A hand wavy explanation of conductivity would be to say the electrons in conductors like aluminum and copper are "free to move around." Solid state physics allows for a more precise understanding of the phenomenon. Using quantum mechanical models we can determine the energy levels that electrons can occupy, and predict how many electrons will be available to conduct electricity.

Semiconductors are the most interesting type of material since they can switch between conductive and non-conductive states. The *transistor*, the magical invention that makes all electronics possible, consists of a sandwich of three different types of semiconductors. The voltage applied to the middle section of a transistor is called the *gate voltage*, and it controls the amount of current that can flow through the transistor. If the gate voltage is set to `ON` (think 1 in binary) then semiconducting material is biased so free electrons are available in its conduction band and current can flow through. If the gate voltage is set to `OFF` (think 0 in binary) then conduction band is depleted and the transistor won't conduct electricity. The improvements in semiconductor technologies, specifically the ability to pack billions of transistors into a tiny microprocessor chip is behind the computers revolution that's been ongoing, pretty much since the transistors were first commercialized. In summary, no solid state physics, no mobile phones.

Quantum mechanics is used in all areas of solid state physics, in

fact we could even say "applied quantum physics" would be another suitable label for the field.

[Simple explanation of the energy band structure and conductivity]
`https://wikipedia.org/wiki/Electrical_resistivity_and_conductivity`

Superconductors

Certain materials exhibit surprising physical properties at very low temperature. When we say "low" temperatures, we mean *really low* like $-272°$C. You'd exhibit surprising properties too if your were placed in an environment this cold! For example there are regular conductors with small resistance, and high-end conductors which have less resistance, and then there are *superconductors* which have zero resistance. Superconductors are an example of a purely quantum phenomenon that cannot be explained by classical physics.

Some of the most iconic landmarks of modern scientific progress like magnetic resonance imaging (MRI) machines, and magnetically levitating bullet trains are made possible by superconductor technology. Superconductors offer zero resistance to electric current, which means they can support much stronger currents than regular conductors like copper and aluminum.

[Superconductivity]
`https://en.wikipedia.org/wiki/Superconductivity`

Quantum optics

Classical optics deals with beams of light that contain quintillions of photons. A *quintillion* is 10^{18}, which is a lot. For experiments with this many photons, it is possible to model light beams as a continuous electromagnetic waves and use classical electromagnetic theory and optics to understand experiments. Quantum optics comes into play when we perform optics experiments using much fewer photons, including experiments that involve single photons. When a single photon travels through an optical circuit it cannot "split" like a continuous wave. For example, when a beam of light hits a half-silvered mirror, we say the beam is "partially" reflected. We can't say the same thing for a single photon since the photon cannot be split. Instead the photon goes into a superposition of "passed through" and "reflected" states, as shown in Figure 10.11 (page 436).

An example of quantum optics effect is the *spontaneous downconversion effect* in which a single photon is absorbed by a material and then reemitted as two photons whose polarization state is entangled:

$$|\Psi_-\rangle = \tfrac{1}{\sqrt{2}}|H\rangle|V\rangle - \tfrac{1}{\sqrt{2}}|V\rangle|H\rangle.$$

Because of the properties of the crystal, we know one of the two emitted photons will have horizontal polarization and the other will have vertical polarization, but we don't know which is which. Such entangled photons can be used as a starting point for other experiments that use entanglement. Another interesting aspect of quantum optics are the so called *squeezed states* that can be detected more accurately than regular (unsqueezed) photons.

Quantum optics is a field of active research. Scientists in academia and industry study exotic photon generation, advanced photon detection schemes, and in general explore how photons can be used most efficiently to carry information.

[Basic principles in physics of light]
`https://materialford.wordpress.com/introduction-to-research-light/`

Quantum cryptography

Performing a quantum measurement on the state $|\psi\rangle$ tends to disturb the state. From the point of view of experimental physics, this is an obstacle since it gives us a limited one-time access to the quantum state $|\psi\rangle$, making studying quantum states more difficult. From the point of view of cryptography however, the state-disturbing aspect of quantum measurement is an interesting property. If Alice transmit a secret message to Bob encoded in the state of a quantum system, then it would be impossible for an eavesdropper Eve to "listen in" on the state unnoticed because Eve's measurement will disturb the state. The *BB84 protocol*, named after its inventors Charles Bennett and Gilles Brassard, is based on this principle.

The standard basis $B_s = \{|0\rangle, |1\rangle\}$ and the Hadamard basis $B_h = \{|+\rangle, |-\rangle\}$ are a *mutually unbiased bases*, which means basis vector from one basis lie exactly "in between" the vectors from the other basis. The use of mutually unbiased bases is central to the security of the BB84 protocol, which we'll describe in point form:

1. Alice transmits $2n$ "candidate" bits of secret key to Bob. She chooses one of the bases B_s or B_h at random when encoding each bit of information she wants to send. Bob chooses to perform his measurement randomly, either in the standard basis or in the Hadamard basis. The information will be transmitted faithfully whenever Bob happens to pick the same basis as Alice, which happens about half the time. If the basis used in Bob' measurement is different from the basis used by Alice for the encoding, Bob's output will be completely random.

2. Alice and Bob publicly announce the basis they used for each transmission and discard the bits where different bases were

used. This leaves Alice and Bob with roughly n candidate bits of secret key.

3. Alice and Bob then publicly reveal αn of the candidate bits, which we'll call the *check bits*. Assuming the quantum communication channel between Alice and Bob does not introduce any noise, Alice and Bob's copies of the check bits should be identical, since the same basis was used. If too many of these *check bits* disagree, they abort the protocol.

4. If the αn check bits agree, then Alice and Bob can be sure the remaining $(1-\alpha)n$ bits they share are only known to them.

Consider what happens if the eavesdropper Eve tries introduce herself between Alice and Bob by measuring the quantum state $|\psi\rangle$ send by Alice, then forwarding to Bob the post measurement state $|\psi'\rangle$. Eve is forced to choose a basis for the measurement she performs and her measurement will disturb the state $|\psi\rangle$ whenever she picks a basis different from the one used by Alice. Since $|\psi'\rangle \neq |\psi\rangle$ Alice and Bob will be able to detect this eavesdropping has occurred because some of the *check bits* they compare in Step 3 will disagree. The BB84 does not use quantum mechanics to prevent eavesdropping, but rather gives Alice and Bob the ability to detect when an eavesdropper is present.

The BB84 protocol established the beginning of a new field at the intersection of computer science and physics which studies *quantum key distribution* protocols. The field has developed rapidly from theory, to research, and today there are even commercial quantum cryptography systems. It's interesting to compare the quantum crypto with the public key cryptography systems discussed in Section 8.9. The security of the RSA public-key encryption is based on the computational difficulty of factoring large numbers. The security of quantum cryptography is guaranteed by the laws of quantum mechanics.

[Bennett–Brassard quantum cryptography protocol from 1984]
`en.wikipedia.org/wiki/BB84`

[Using quantum phenomena to distribute secret key]
`https://en.wikipedia.org/wiki/Quantum_key_distribution`

Quantum computing

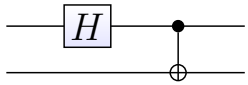
The idea of quantum computing—in one form or another—has been around since the early days of quantum physics. Richard Feynman originally proposed the idea of a *quantum simulator*, which is a quantum apparatus that can simulate the quantum behaviour of another physical system. Imagine a physical system that is difficult and expensive to build, but whose behaviour can be simulated on the quantum

simulator. A quantum simulator will be much better at simulating quantum phenomena than any simulation of quantum physics on a classical computer.

Another possible application of a quantum simulator would be to encode classical mathematical optimization problems as constraints in a quantum system, then use let the quantum evolution of the system "search" for good solutions. Using a quantum simulator in this way, it might be possible to find solutions to a optimization problems much faster than any classical optimization algorithm would able to.

Once computer scientists started thinking about quantum computing, they were not satisfied just with studying only optimization problems, but set out to qualify and quantify all the computational tasks that are possible with qubits. A quantum computer stores and manipulates information that is encoded as quantum states. It is possible to perform certain computational tasks on in the quantum world much faster than on any classical computer.

Quantum circuits Computer scientists like to think quantum computing tasks as series of "quantum gates," in analogy with the logic gates used to construct classical computers. Figure 10.20 shows an example of a quantum circuit that takes two qubits as inputs and produces two qubits as outputs.



$$\Leftrightarrow \quad \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

Figure 10.20: A quantum circuit that applies the Hadamard gate to the first qubit then applies the control-NOT gate from the first qubit to the second qubit.

This circuit in Figure 10.20 is the combination of two quantum gates. The first operation is to apply the Hadamard gate H on the first qubit, leaving the second qubit untouched. This operation is equivalent to multiplying the input state by the matrix $H \otimes \mathbb{1}$. The second operation is called the *control-NOT* (or control-X) gate, which applies the X operator (also known as the NOT gate) to the second qubit whenever the first qubit is $|1\rangle$, and does nothing otherwise:

$$\text{CNOT}(|0\rangle \otimes |\varphi\rangle) = |0\rangle \otimes |\varphi\rangle, \qquad \text{CNOT}(|1\rangle \otimes |\varphi\rangle) = |1\rangle \otimes X|\varphi\rangle.$$

The circuit illustrated in Figure 10.20 can be used to create entangled quantum states. If we input the quantum state $|00\rangle \equiv |0\rangle \otimes |0\rangle$ into the circuit, we obtain the maximally entangled state $|\Phi_+\rangle \equiv \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ as output:

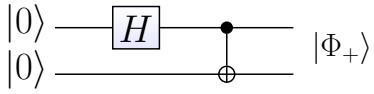


Figure 10.21: Inputting $|0\rangle \otimes |0\rangle$ into the circuit produces an EPR state $|\Phi_+\rangle \equiv \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ on the two output wires of the circuit.

Quantum measurements can also be represented in quantum circuits. Figure 10.22 shows how a quantum measurement in the standard basis is represented.

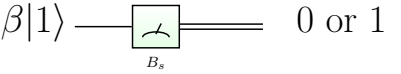


Figure 10.22: Measurement in the standard basis $B_s = \{|0\rangle, |1\rangle\}$. The projectors of this measurement are $\Pi_0 = |0\rangle\langle 0|$ and $\Pi_1 = |1\rangle\langle 1|$.

We use double lines to represent the flow of classical information in the circuit.

Quantum registers Consider a quantum computer with a single register $|R\rangle$ that consists of three qubits. The quantum state of this quantum register is a vector in $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$:

$$|R\rangle = \ (\alpha_1|0\rangle + \beta_1|1\rangle) \ \otimes \ (\alpha_2|0\rangle + \beta_2|1\rangle) \ \otimes \ (\alpha_3|0\rangle + \beta_3|1\rangle)\,,$$

where the tensor product $\otimes$ is used to combine the quantum states of the individual qubits. We'll call this the "physical representation" of the register and use 0-based indexing for the qubits. Borrowing language from classical computing, we'll call the rightmost qubit the *least significant* qubit, and the leftmost qubit the *most significant* qubit.

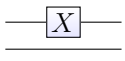
The tensor product of three vectors with dimension two is a vector with dimension eight. The quantum register $|R\rangle$ is thus a vector in an eight-dimensional vector space. The quantum state of a three-qubit register can be written as:

$$|R\rangle = a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle + a_4|4\rangle + a_5|5\rangle + a_6|6\rangle + a_7|7\rangle,$$

where a_i are complex coefficients. We'll call this eight-dimensional vector space the "logical representation" of the quantum register. Part of the excitement about quantum computing is the huge size of the "logical space" where quantum computations take place. The logical space of a 10-qubit quantum register has dimension $2^{10} = 1024$. That's 1024 complex coefficients we're talking about. That's a big state space for just a ten-qubit quantum register. Compare this with a 10-bit classical register which can store just one of $2^{10} = 1024$ discrete values.

We cannot discuss quantum computing further but I still want to show you some examples of single-qubit quantum operations and their effect on the tensor product space, so you'll have an idea of what kind of craziness is possible.

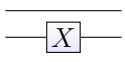
Quantum gates Let's say you've managed to construct a quantum register, what can you do with it? Recall the single-qubit quantum operations Z, X, and H we described earlier. We can apply any of these operations on individual qubits in the quantum register. For example applying the $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ gate to the first (most significant) qubit of the quantum register corresponds to the following quantum operation:



$$\Leftrightarrow \qquad X \otimes \mathbb{1} \otimes \mathbb{1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The operator $X \otimes \mathbb{1} \otimes \mathbb{1}$ "toggles" the first qubit in the register while leaving all other qubits unchanged.

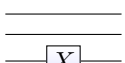
Yes, I know the tensor product operation is a bit crazy, but that's the representation of composite quantum systems and operations so you'll have to get used to it. What if we apply the X operator applied on the middle qubit?



$$\Leftrightarrow \qquad \mathbb{1} \otimes X \otimes \mathbb{1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Compare the structure of the operators $X \otimes \mathbb{1} \otimes \mathbb{1}$ and $\mathbb{1} \otimes X \otimes \mathbb{1}$. Do you see how the action of X affect different dimensions in the tensor product space $\mathbb{C}^8$?

To complete the picture, let's also see the effects of applying the X gate to the third (least significant) qubit in the register:



$$\Leftrightarrow \qquad \mathbb{1} \otimes \mathbb{1} \otimes X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Crazy stuff, right? Don't worry you'll get used to the whole space-within-a-space structure with time.

Okay so what?

We discussed quantum registers and quantum operations, but we still haven't said what quantum computing is good for, if anything. Quantum computers give us access to a very large state space. The fundamental promise of quantum computing is that a small set of simple quantum operations (quantum gates) can be used to perform interesting computational tasks. Sure it's difficult to interact with quantum systems (state preparation and measurement), but damn the space is big so it's worth checking out what kind of computing you can do in there. It turns out there are already several useful things you can do using a quantum computer. The two flagship applications for quantum computing are Grover's search algorithm and Shor's factoring algorithm.

Grover's search algorithm Suppose you're given an unsorted list of n items and you want to find a particular item in that list. This is called and *unstructured search problem*. This is a hard problem to solve for a classical computer since the algorithm would have to go through the entire list, which will take on the order of n steps. In contrast, the unstructured problem can be solved in roughly $\sqrt{n}$ steps on a quantum computer using *Grover's algorithm*.

This "quantum speedup" for doing unstructured search is a nice-to-have, but the real money maker for the field of quantum computing has been Shor's factoring algorithm for factoring biprime numbers.

Shor's factoring algorithm The security of the RSA crypto system we discussed in Section 8.9 is based on the assumption that factoring large *biprime* numbers is computationally intractable. Given the product de of two unknown prime numbers d and e, it is computationally difficult to find the factors e and d. No classical algorithm is known that can factor large numbers; even the letter agencies will have a hard time finding the factors of de when d and e are chosen to be sufficiently large prime numbers. Thus, if an algorithm that could quickly factor large numbers existed, attackers would be able to break many of current security systems. *Shor's factoring algorithms* fits the bill, theoretically speaking.

Shor's algorithm reduces the factoring problem to the problem of *period finding*, which can be solved efficiently using the quantum Fourier transform. Shor's algorithm can factor large numbers efficiently (in polynomial time). This means RSA encryption would be easily "hackable" using Shor's algorithm running on sufficiently large, and sufficiently reliable quantum computer. The letter agencies really got really excited about this development since they'd love to be able to hack all of present-day cryptography. Can you imagine not being able to login securely to any website because Eve is listening in and hacking your crypto using her quantum computer?

Shor's algorithms is currently only a *theoretical* concern. Despite considerable effort, no quantum computers exist today that would allow us to manipulate quantum registers with thousands of qubits. It seems that the letter agencies didn't get the memo on this, and they think that quantum computers will be easy to build. Perhaps quantum researchers funded by various military agencies deserve a Nobel peace prize for all the funds they have diverted from actual weapons research and into research on fundamental science.

Discussion

Quantum computing certainly opens some interesting possibilities, but we shouldn't get ahead of ourselves and imagine a quantum computing revolution is just around the corner. As with startup ideas, the implementation is what counts—not the idea. The current status of quantum computing as a technology is mixed. On one hand certain quantum algorithms performed in logical space are very powerful; on the other hand the difficulty of building a quantum computer is not to be missunderestimated.

It's also important to keep in mind that quantum computers are not better at solving arbitrary computational problems. The problems for which there exists a quantum speedup have a particular structure, which can be tackled with a choreographed pattern of constructive and

destructive interference in quantum registers. Not all computationally hard problems have this structure. Quantum computing technology is at a cross road: it could turn out to be a revolutionary development, or it could turn out that building a large-scale quantum computer is too much of an engineering challenge. Basically, don't go out thinking quantum is the big new thing. It's cool that we can do certain tasks faster on a quantum computer, but don't throw out classical computer just yet.

Even if the quest to build a quantum computer doesn't pan out in the end, we'll have learned many interesting things about fundamental physics along the way. Indeed learning about the fundamental nature quantum information is more scientifically valuable than focussing on how to hack into people's email. In the next section we'll discuss an example of a fundamental results of quantum information science.

Quantum teleportation Figure 10.23 illustrates one of the surprising aspects of quantum information: we can "teleport" a quantum state $|\psi\rangle$ from one lab to another using one maximally entangled state shared between Alice's and Bob's labs and two bits of classical communication from Alice to Bob. The quantum state $|\psi\rangle$ starts off in the first qubit of the register, which we assume is held by Alice, and ends up in the third qubit which is Bob's lab. We can express the the quantum teleportations protocol as a quantum circuit.

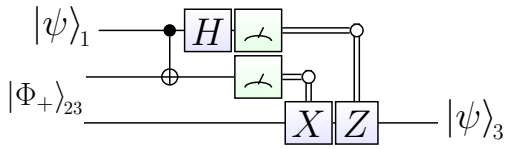


Figure 10.23: The first two qubits are assumed to be in Alice's lab. The state of the first qubit $|\psi\rangle_1$ is transferred into the third qubit $|\psi\rangle_3$, which is held by Bob. We say ψ is "teleported" from Alice's lab to Bob's lab because only classical information was used to transfer the state. The information about the results of the two measurements are classical bits, which are easy to send over.

The quantum teleportation protocol requires that Alice and Bob preshare a maximally-entangled state $|\Phi_+\rangle$. This could be achieved if Alice and Bob get together in a central location and produce an entangled state using the circuit shown in Figure 10.21. Alice and Bob then bring their respective halves of the entangled state to their labs. Note Bob's lab could be very far away from Alice's lab, in another building, in another city, or even at the other end of the world.

The initial state for the quantum teleportation protocol is

$$|\psi\rangle_1 \otimes |\Phi_+\rangle_{23} = \Big(\alpha|0\rangle_1 + \beta|1\rangle_1\Big) \otimes \Big(\tfrac{1}{\sqrt{2}}|00\rangle_{23} + \tfrac{1}{\sqrt{2}}|11\rangle_{23}\Big).$$

Alice has two qubits in her lab, the state $|\psi\rangle_1 = \alpha|0\rangle_1 + \beta|1\rangle_1$ and half of the entangled state, Bob has the third qubit, which is the other half of the entangled state. At the end of the teleportation protocol the information about the state ψ will appear in Bob's lab: $|\psi\rangle_3 = \alpha|0\rangle_3 + \beta|1\rangle_3$.

Without quantum communication it seems impossible for Alice to communicate the coefficients α and β to Bob. The pre-shared entanglement between Alice and Bob somehow enables this feat. As soon as Alice performs the measurement of her two qubits, the quantum information about the sate ψ becomes available in Bob's lab, up to a "correction factor" of $\mathbb{1}$, X, Z, or X followed by Z, that Bob has to apply on the third qubit. Bob will not have the state information until he learns which of the four recovery operations he must perform. The "which correction" information can be transmitted by classical means: Alice can shout the result to Bob if he's next door, tell him the results on the phone, or send him a text message. After applying the needed recovery gate(s), Bob ends up with the state $|\psi\rangle_3 = \alpha|0\rangle_3 + \beta|1\rangle_3$.

We can describe this protocol as "quantum teleportation" because Alice seems to "beam up" the state to Bob by sending him only classical information. That's true, but the more interesting part is the pre-shared entanglement they used, which really makes this work. I have complained enough about science journalists by this point in the chapter, so I won't repeat myself, but you can imagine they had a field trip with this one.

The need for pre-shared entanglement $|\Phi_+\rangle$ between Alice and Bob is analogous to how Alice and Bob needed to pre-share a secret key $\vec{k}$ in order to use the one-time pad encryption protocol. Indeed, pre-shared entangled states are a prime resource in quantum information science. The *superdense coding protocol* is another surprising application of quantum entanglement. Using this protocol Alice can communicate two bits of classical information to Bob by sending him a single qubit and consuming one pre-shared entangled state.

Links

[Quantum simulators and practical implementations]
`https://en.wikipedia.org/wiki/Quantum_simulator`

[Some data about the difficulty of RSA factoring]
`https://en.wikipedia.org/wiki/RSA_numbers`

[An introduction to quantum computing]
http://arxiv.org/abs/0708.0261v1/

[Video tutorials on quantum computing by Michael Nielsen]
http://michaelnielsen.org/blog/quantum-computing-for-the-determined/

[Grover's algorithm for unstructured search]
https://en.wikipedia.org/wiki/Grover's_algorithm

[Shor's algorithm for factoring biprime integers]
https://en.wikipedia.org/wiki/Shor's_algorithm

Quantum error correcting codes

Quantum states are finicky things. Every interaction that a qubit has with its environment corrupts the quantum information that it stores. In the previous section we talked about quantum computing in the abstract, assuming the existence of an ideal noiseless quantum computer. The real world is a noisy place, so constructing a practical quantum computer in the presence of noise is a much more difficult challenge.

Recall that errors caused by noise were also a problem for classical computers, and we solved that problem using error correcting codes. Can we use error correcting codes on quantum computers too? Indeed it's possible to use *quantum error correcting codes* to defend against the effect of quantum noise. Keep in mind, quantum error correcting codes are more complicated to build than their classical counterparts, so it's not an "obvious" thing to do, but it can be done.

We don't want to go into too much details, but it's worth pointing out the following interesting fact about quantum error correction. Building quantum error correcting codes that can defend agains a finite set of errors is sufficient to defend against all possible types of errors. The use of quantum error correcting schemes is analogous to the classical error correcting schemes we saw in Section 8.10. We encode k qubits of data that we want to protect from noise into a larger n-qubit state. The encoded state can support some number of errors before losing the data. The error correction procedure involves a syndrome measurement on a portion of the state, and "correction" operators applied to the remaining part. I encourage your to follow the links provided below to learn more about the topic.

Building reliable "quantum gates" is a formidably difficult task because of the fundamental difficulty of protecting qubits from noise, and at the same time enabling quantum operations and strong interactions between qubits. It is the author's opinion that Feynman's original idea of building quantum simulators for physical systems will be the first useful applications in quantum computing.

[More on quantum error correcting codes]
`https://en.wikipedia.org/wiki/Quantum_error_correction`

Quantum information theory

Classical information theory studies problems like the compression of information and the transmission of information through noisy communication channels. Quantum information theory studies the analogous problems of compression of quantum information and the communication over noisy quantum channels.

The appearance of the word "theory" in "quantum information theory" should give you a hint that this is mostly a theoretical area of research in which problems are studied in the abstract. The main results in information theory are abstract theorems that may not have direct bearing on practical communication scenarios. Applications of quantum information theory are still far into the future, but that's how it is with theory subjects in general. The classical information theory theorems proved in the 1970s probably looked like "useless theory" too, but these theorems serve as the basis of all modern wireless communications. Perhaps 40 years form now, the purely theoretical quantum information theorems of today will be used to solve the practical communication problems of the future.

Current efforts in quantum information theory aim to established capacity results for quantum channels. Some of the existing results are directly analogous to classical capacity results. Other problems in quantum information theory, like the use of entanglement-assisted codes, have no classical counterparts and require completely new way of thinking about communication problems. The book by Wilde is an excellent guide to the field.

Recently quantum theory has been applied to novel communication systems, and there is a growing interest from industry to develop applications that push optical communication channels to their theoretical efficiency bounds. Essentially, quantum networks are being invented in parallel with quantum computers, so when we finally build quantum computers, we'll be able to connect them together, presumably so they can share funny cat videos. What else?

[BOOK] Mark M. Wilde. *From Classical to Quantum Shannon Theory*, Cambridge, ISBN 1107034256, `arxiv.org/abs/1106.1445`.

Conclusion

Thank you for sticking around until the end. I hope this chapter helped you see the basic principles of quantum mechanics and demystified some of the sensationalist aspects of the quantum world.

There's nothing too counterintuitive in quantum mechanics, it's just linear algebra, right? With this chapter, I wanted to bring you into the fold about this fascinating subject. Above all, I hope you'll never let any science journalist ever mystify you with a hand-wavy quantum mechanics story. I also expect you to call bullshit on any writer that claims quantum mechanics is related to your thought patterns, or—more ridiculously still—that your thoughts *cause* reality to exist. It could be, but prove it!

I would like to conclude with an optimistic outcome about what is to come. A couple of hundred years ago quantum mechanics was seen as a foreign thing not to be trusted, but with time physicists developed good models, found better ways to explain things, wrote good books, so that now the subject is taught to undergraduate physics students. This gives me hope for humanity that we can handle even the most complex and uncertain topics, when we put our minds into it.

Today we face many complex problems like consolidated corporate control of innovation, cartels, corruption, eroding democratic government systems, the militarization of everything, and conflicts of ideologies. We have Sunni and Shia brothers fighting each other for no good reason. We have all kinds of other bullshit divisions between us. Let's hope that one hundred years from now we have a firm hold on these aspects of human nature, so we can realize the potential of every child, born anywhere in the world.

10.9 Quantum mechanics problems

P10.1 You work in a quantum computing startup and your boss asks you to implement the quantum gate $Q = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$. Can you do it?

Hint: Recall the requirements for quantum gates.

P10.2 The Hadamard gate is defined as $H = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. Compute the effect of the operator HH on the elements of the standard basis $\{|0\rangle, |1\rangle\}$.

P10.3 Specifying an arbitrary vector $\alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$ requires four parameters: the real and imaginary parts of α and β. Thus one might think that quantum states in $\mathbb{C}^2$ have four *degrees of freedom*. However, the unit-length requirement and the fact that global phase of a qubit can be ignored correspond to additional constraints that reduce the number of degrees of freedom. How many parameters are required to specify a general quantum state $|\psi\rangle \in \mathbb{C}^2$?

P10.4 We can write any qubit using only two real parameters

$$|\psi\rangle = \alpha|0\rangle + \sqrt{1-\alpha^2}e^{i\varphi}|1\rangle,$$

where $\alpha \in \mathbb{R}$ and $\varphi \in \mathbb{R}$. What are the ranges of values for α and φ such that all qubits can be represented?

P10.5 Another choice of parametrization for qubits is to use two angles θ and φ:

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,e^{i\varphi}|1\rangle.$$

What are the ranges of values for θ and φ such that all qubits can be represented?

P10.6 Calculate the joint quantum state of the three qubits at each step in the quantum teleportation circuit in Figure 10.23. Denote $|R_0\rangle = (\alpha|0\rangle + \beta|1\rangle)_1 \otimes |\Phi_+\rangle_{23}$ the initial state, $|R_1\rangle$ the state after the control swap gate, $|R_2\rangle$ after the H gate, and the final state after the correction operator has been applied is $|R_3\rangle$.

P10.7 Consider a one-dimensional model of a particle caught between two ideal mirrors. If we assume the two mirrors are placed at a distance of 1 unit apart, the wave function description of the system is $\psi(x)$, where $x \in [0, 1]$. Find the probability of observing x between 0 and 0.1 for the following wave functions: (a) Uniform distribution on $[0, 1]$, (b) $\psi_b(x) = 2x - 1$, (c) $\psi_c(x) = 6x^2 - 6x + 1$.

P10.8 Show that the functions $\psi_b(x) = 2x - 1$ and $\psi_c(x) = 6x^2 - 6x + 1$ are orthogonal with respect to the inner product $\langle f, g\rangle = \int_0^1 \overline{f(x)}g(x)\,dx$.

End matter

Conclusion

By surviving the linear algebra math covered in this book you've proven you can handle abstraction. Good job! That's precisely the skill needed for understanding more advanced math concepts, building scientific models, and profiting from useful applications. Understanding concepts like dimensions, orthogonality, and length in abstract vector spaces is very useful. Congratulations on your first steps towards mathematical enlightenment.

Let's review where we stand in terms of math modelling tools. The first step when learning math modelling is to understand basic math concepts such as numbers, equations, and functions $f : \mathbb{R} \to \mathbb{R}$. Once you know about functions, you can start to use different formulas $f(x)$ to represent, model, and predict the values of real-world quantities. Using functions is the first modelling superpower conferred on people who become knowledgeable in math. Mathematical models emerge as a naturally-useful common core for all of science. For example, by understanding the properties of the function $f(x) = Ae^{-x/B}$ in the abstract will enable you to describe the expected number of atoms remaining in a radioactive reaction $N(t) = N_o e^{-\gamma t}$, predict the voltage of a discharging capacitor over time $v(t) = V_o e^{-\frac{t}{RC}}$, and understand the exponential probability distribution $p_X(x) = \lambda e^{-\lambda x}$.

The second step toward math modelling enlightenment is to generalize the inputs x, the outputs y, and functions f to other contexts. In linear algebra, we studied functions $T : \mathbb{R}^n \to \mathbb{R}^m$, which are linear:

$$T(\alpha\vec{x}_1 + \beta\vec{x}_2) = \alpha T(\vec{x}_1) + \beta T(\vec{x}_2).$$

The linear property enables us to calculate things, solve equations involving linear transformations, and build models with interesting structure (recall the applications discussed in Chapter 8). The mathematical structure of a linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ is faithfully represented as a multiplication by a matrix $M_T \in \mathbb{R}^{m \times n}$. The notion of matrix *representations* ($T \Leftrightarrow M_T$) played a central role in

this book. Seeing the parallels and similarities between the abstract math notion of a linear transformation and its concrete representation as a matrix is essential for the second step of math enlightenment.

If you didn't skip the sections on abstract vector spaces, you also know about the parallels between the vector space $\mathbb{R}^4$, and the abstract vector spaces of third-degree polynomials $a_0+a_1x+a_2x^2+a_3x^3$ and 2×2 matrices $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$. This is another step on your way up the ladder of abstraction, though I don't think it counts as a third step since there is no new math—just some basic observations and cataloguing of math objects that have vector structure.

The computational skills you learned in Chapter 4 are also useful, but you probably won't be solving any problems by hand using row operations from this point forward, since computers outclass you many-to-one on matrix arithmetic tasks. Good riddance. Until now you did all the work and used `SymPy` to check your answers, from now on you can let `SymPy` do all the calculations and your job will be to chill.

It was a great pleasure for me to be your guide through the topics of linear algebra. I hope you walk away from this book with a solid understanding of how the concepts of linear algebra fit together. In the introduction to the book, I likened linear algebra to playing with LEGOs. Indeed, if you feel comfortable with manipulating vectors and matrices, performing change-of-basis operations, and using the matrix decomposition techniques to see inside matrices, you'll be able to "play" with all kinds of complex systems and problems. For example, consider the linear transformation T that you want to apply to an input vector $\vec{v}$. Suppose the linear transformation T is most easily described in the basis B', but the vector $\vec{v}$ is expressed with respect to the basis B. "No problem," you say, and proceed straight away to build the following chain of matrices that will compute the output vector $\vec{w}$:

$$[\vec{w}]_B = {}_B[\mathbb{1}]_{B'}\,{}_{B'}[A_T]_{B'}\,{}_{B'}[\mathbb{1}]_B[\vec{v}]_B.$$

Do you see how matrices and vectors fit together neatly like LEGOs?

I can't tell what the next step on your journey will be. With linear algebra modelling skills under your belt, you have a thousand doors open to you, and you must explore and choose. Will you go on to learn how to code and start a software company? Maybe you can use your analytical skills to go to wall street and destroy the System from the inside? Or perhaps you'll apply your modelling skills to revolutionize energy generation, thus making human progress sustainable. Regardless of your choice of career, I want you to stay on good terms with math and keep learning when you have a chance. Good luck with your studies!

Social stuff

Be sure to send me feedback if you liked or hated this book. Feedback is crucial so I know how to adjust the writing, the content, and the attitude of the book for future readers. I'm sure you found some parts of the book that were not explained well or otherwise weak. Please take the time to drop me a line and let me know. This way I'll know which sections to fix in the next revision. You can reach me by email at `ivan.savov@gmail.com`.

Another thing you could do to help me is to write a review of the book on amazon.com, goodreads, google books, or otherwise spread the word about the NO BULLSHIT GUIDE series of books. Talk to your friends and let them in on the math buzz.

If you want to know what `Minireference Co.` has been up to, check out the company blog at `minireference.com/blog/`. The blog is a mix of 30% technology talk, 50% book business talk, and 20% announcements. Checking the blog is the easiest way to check the progress of the revolution in the textbook industry. You can also connect via twitter `@minireference` and facebook `fb.me/noBSguide`.

Acknowledgements

This book would not have been possible without the help of my parents, and my teachers: Paul Kenton, Benoit Larose, Ioannis Psaromiligkos, and Patrick Hayden. It's from them that I learned to trick students into learning advanced material. Many thanks also to David Avis, Arlo Breault, Michael Hilke, Igor Khavkine, Juan Pablo Di Lelle, Ivo Panayotov, and Mark M. Wilde for their support with the book.

The errorlessness and consistency of the text would not have been possible without the help of my editor Sandy Gordon, who did a great job at polishing the text until it flowed. Truly no bullshit is allowed into the book when Sandy's on watch. Many thanks to Polina Anis'kina who helped me to create the problem sets for the book.

General linear algebra links

Below are some useful links for you to learn more about linear algebra. We covered a lot of ground, but linear algebra is endless. Don't sit on your laurels now having completed this book and the problem sets and think you're the boss. You have all the tools, but you need to practice using them. Try reading up on the same topics from some other sources. See if you can do the problem sets in another linear

algebra textbook. Try to make some use of linear algebra in the coming year to solidify your understanding of the material.

[Video lectures of Gilbert Strang's Linear algebra class at MIT]
http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010

[Lecture notes by Terrence Tao]
http://www.math.ucla.edu/~tao/resource/general/115a.3.02f/

[Wikipedia overview on matrices]
https://en.wikipedia.org/wiki/Matrix_(mathematics)

[Linear algebra wikibook (with solved problems)]
https://en.wikibooks.org/wiki/Linear_Algebra

[Proofs involving linear algebra]
http://proofwiki.org/wiki/Category:Linear_Transformations
http://proofwiki.org/wiki/Category:Linear_Algebra
http://proofwiki.org/wiki/Category:Matrix_Algebra

Appendix A

Answers and solutions

Chapter 1 solutions

Answers to exercises

E1.1 $x = 2$, $y = 3$. **E1.2** $x = 5$, $y = 6$, and $z = -3$. **E1.3** $p = 7$ and $q = 3$. **E1.4** $x = 2$, $y = 1$.

Answers to problems

P1.1 $x = \pm 4$. **P1.2** $x = A\cos(\omega t + \phi)$. **P1.3** $x = \frac{ab}{a+b}$. **P1.4** (1) 2.2795. (2) 1024. (3) −8.373. (4) 11. **P1.5** (1) $\frac{3}{4}$. (2) $\frac{-141}{35}$. (3) $3\frac{23}{32}$. **P1.6** (1) c. (2) 1. (3) $\frac{9a}{b}$. (4) a. (5) $\frac{b}{ac}$. (6) $x^2 + ab$. **P1.7** (1) $x^2 + (a-b)x - ab$. (2) $2x^2 - 7x - 15$. (3) $10x^2 + 31x - 14$. **P1.8** (1) $(x-4)(x+2)$. (2) $3x(x-3)(x+3)$. (3) $(x+3)(6x-7)$. **P1.9** (1) $(x-2)^2 + 3$. (2) $2(x+3)^2 + 4$. (3) $6\big(x + \frac{11}{12}\big)^2 - \frac{625}{24}$. **P1.10** \$0.05. **P1.11** 13 people, 30 animals. **P1.12** 5 years later. **P1.13** girl = 80 nuts, boy = 40 nuts. **P1.14** Alice is 15. **P1.15** 18 days. **P1.16** After 2 hours. **P1.18** $\varphi = \frac{1+\sqrt{5}}{2}$. **P1.19** $x = \frac{-5 \pm \sqrt{41}}{2}$. **P1.20** (1) $x = \sqrt[3]{2}$. (2) $x = (\frac{\pi}{2} + 2\pi n)$ for $n \in \mathbb{Z}$. **P1.21** No real solutions if $0 < m < 8$. **P1.22** (1) e^z. (2) $\frac{x^3 y^{15}}{z^3}$. (3) $\frac{1}{4x^4}$. (4) $\frac{1}{4}$. (5) -3. (6) $\ln(x+1)$. **P1.23** $\epsilon = 1.110 \times 10^{-16}$; $n = 15.95$ in decimal. **P1.24** (1) $x \in (4, \infty$. (2) $x \in [3, 6]$. (3) $x \in -\infty, -1] \cup [\frac{1}{2}, \infty$. **P1.25** For $n > 250$, Algorithm Q is faster. **P1.26** 10 cm. **P1.27** 22.52 in. **P1.28** $h = \sqrt{3.33^2 - 1.44^2} = 3\,\text{m}$. **P1.29** The opposite side has length 1. **P1.30** $x = \sqrt{3}$, $y = 1$, and $z = 2$. **P1.31** $d = \frac{1800\tan 20^\circ - 800\tan 25^\circ}{\tan 25^\circ - \tan 20^\circ}$, $h = 1658.46\,\text{m}$. **P1.32** $x = \frac{2000}{\tan 24^\circ}$. **P1.33** $x = \tan\theta\sqrt{a^2 + b^2 + c^2}$. **P1.34** $a = \sqrt{3}$, $A_\triangle = \frac{3\sqrt{3}}{4}$. **P1.35** $\sin^2\theta\cos^2\theta = \frac{1-\cos 4\theta}{8}$. **P1.36** $P_\bigcirc = 16\tan(22.5^\circ)$, $A_\bigcirc = 8\tan(22.5^\circ)$. **P1.37** $c = \frac{a\sin 75^\circ}{\sin 41^\circ} \approx 14.7$. **P1.38** (a) $h = a\sin\theta$. (b) $A = \frac{1}{2}ba\sin\theta$. (c) $c = \sqrt{a^2 + b^2 - 2ab\cos(180 - \theta)}$. **P1.39** $B = 44.8^\circ$, $C = 110.2^\circ$. $c = \frac{a\sin 110.2^\circ}{\sin 25^\circ} \approx 39.97$. **P1.40** $v = 742.92$ km/h. **P1.41** 1.33 cm. **P1.42** $x = 9.55$. **P1.43** $\frac{1}{2}(\pi 4^2 - \pi 2^2) = 18.85\ \text{cm}^2$. **P1.44** $\ell_{\text{rope}} = 7.83\,\text{m}$. **P1.45** $A_{\text{rect}} = 5c + 10$. **P1.46** $V_{\text{box}} = 1.639\,\text{L}$. **P1.47** $\theta = 120^\circ$. **P1.48** $\frac{R}{r} = \frac{1-\sin 15^\circ}{\sin 15^\circ} = 2.8637$. **P1.49** 7 cm. **P1.50** $V = 300\,000\,\text{L}$. **P1.51** 315 000 L. **P1.52** 4000 L. **P1.53** $d = \frac{1}{2}(35 - 5\sqrt{21})$. **P1.54** A rope

of length $\sqrt{2}\ell$. **P1.55** 20 L of water. **P1.56** $h = 7.84375$ inches. **P1.57** $1 + 2 + \cdots + 100 = 50 \times 101 = 5050$. **P1.58** $x = -2$ and $y = 2$. **P1.59** $x = 1$, $y = 2$, and $z = 3$. **P1.60** \$112. **P1.61** 20%. **P1.62** \$16501.93. **P1.64** 0.14 s. **P1.65** $\tau = 34.625$ min, 159.45 min. **P1.66** $V(0.01) = 15.58$ volts. $V(0.1) = 1.642$ volts.

Solutions to selected problems

P1.5 For (3), $1\frac{3}{4} + 1\frac{31}{32} = \frac{7}{4} + \frac{63}{32} = \frac{56}{32} + \frac{63}{32} = \frac{119}{32} = 3\frac{23}{32}$.

P1.9 The solutions for (1) and (2) are fairly straightforward. To solve (3), we first factor out 6 from the first two terms to obtain $6(x^2+\frac{11}{6}x)-21$. Next we choose half of the coefficient of the linear term to go inside the square and add the appropriate correction to maintain equality: $6[x^2 + \frac{11}{6}x] - 21 = 6[(x + \frac{11}{12})^2 - \left(\frac{11}{12}\right)^2] - 21$. After expanding the rectangular brackets and simplifying, we obtain the final expression: $6\left(x + \frac{11}{12}\right)^2 - \frac{625}{24}$.

P1.11 Let p denote the number of people and a denote the number of animals. We are told $p + a = 43$ and $a = p + 17$. Substituting the second equation into the first, we find $p + (p + 17) = 43$, which is equivalent to $2p = 26$ or $p = 13$. There are 13 figures of people and 30 figures of animals.

P1.12 We must solve for x in $35 + x = 4(5 + x)$. We obtain $35 + x = 20 + 4x$, then $15 = 3x$, so $x = 5$.

P1.13 Let's consider x =the number of nuts collected by a boy. Therefore $2x$ =the number of nuts collected by a girl. We get $x + 2x = 120$, so $x = 40$. The number of nuts collected by a boy= 40, The number of nuts collected by a girl= 80.

P1.14 Let A be Alice's age and B be Bob's age. We're told $A = B + 5$ and $A + B = 25$. Substituting the first equation into the second we find $(B + 5) + B = 25$, which is the same as $2B = 20$, so Bob is 10 years old. Alice is 15 years old.

P1.15 The first shop can bind $4500/30 = 150$ books per day. The second shop can bind $4500/45 = 100$ books per day. The combined production capacity rate is $150 + 100 = 250$ books per day. It will take $4500/250 = 18$ days to bind the books when the two shops work in parallel.

P1.16 Let x denote the distance the slower plane will travel before the two planes meet. Let t_{meet} denote the time when they meet, as measured from the moment the second plane departs. The slower plane must travel x km in t_{meet} hours, so we have $t_{\text{meet}} = \frac{x}{600}$. The faster plane is 600 km behind when it departs. It must travel a distance $(x + 600)$ km in the same time so $t_{\text{meet}} = \frac{x+600}{900}$. Combining the two equations we find $\frac{x}{600} = \frac{x+600}{900}$. After cross-multiplying we find $900x = 600x + 600^2$, which has solution $x = 1200$ km. The time when the planes meet is $t_{\text{meet}} = 2$ hours after the departure of the second plane.

P1.17 This is a funny nonsensical problem that showed up on a school exam. I'm just checking to make sure you're still here.

P1.21 Using the quadratic formula, we find $x = \frac{m \pm \sqrt{m^2 - 8m}}{4}$. If $m^2 - 8m \geq 0$, the solutions are real. If $m^2 - 8m < 0$, the solutions will be complex numbers. Factoring the expressions and plugging in some numbers, we observe that $m^2 - 8m = m(m - 8) < 0$ for all $m \in (0, 8)$.

P1.23 See `bit.ly/float64prec` for the calculations.

P1.24 For (3), $1\frac{3}{4} + 1\frac{31}{32} = \frac{7}{4} + \frac{63}{32} = \frac{56}{32} + \frac{63}{32} = \frac{119}{32} = 3\frac{23}{32}$.

P1.25 The running time of Algorithm Q grows linearly with the size of the problem, whereas Algorithm P's running time grows quadratically. To find the size of the problem when the algorithms take the same time, we solve $P(n) = Q(n)$, which is $0.002n^2 = 0.5n$. The solution is $n = 250$. For $n > 250$, the linear-time algorithm (Algorithm Q) will take less time.

P1.29 Solve for b in Pythagoras' formula $c^2 = a^2 + b^2$ with $c = \varphi$, and $a = \sqrt{\varphi}$. The triangle with sides 1, $\sqrt{\varphi}$, and φ is called Kepler's triangle.

P1.30 Use Pythagoras' theorem to find x. Then use $\cos(30°) = \frac{\sqrt{3}}{2} = \frac{x}{z}$ to find z. Finally use $\sin(30°) = \frac{1}{2} = \frac{y}{z}$ to find y.

P1.31 Observe the two right-angle triangles drawn in Figure 1.24. From the triangle with angle 25° we know $\tan 25° = \frac{h}{800+d}$. From the triangle with angle 20° we know $\tan 20° = \frac{h}{1800+d}$. We isolate h in both equations and eliminate h by equating $(1800 + d)\tan 25° = \tan 20°(800 + d)$. Solving for d we find $d = \frac{1800\tan 20° - 800\tan 25°}{\tan 25° - \tan 20°} = 2756.57\,\text{m}$. Finally we use $\tan 25° = \frac{h}{800+d}$ again to obtain $h = \tan 25°(800 + d) = 1658.46\,\text{m}$.

P1.32 Consider the right-angle triangle with base x and opposite side 2000. Looking at the diagram we see that $\theta = 24°$. We can then use the relation $\tan 24° = \frac{2000}{x}$ and solve for x.

P1.34 The internal angles of an equilateral triangle are all 60°. Draw three radial lines that connect the centre of the circle to each vertex of the triangle. The equilateral triangle is split into three obtuse triangles with angle measures 30°, 30°, and 120°. Split each of these obtuse sub-triangles down the middle to obtain six right-angle triangles with hypotenuse 1. The side of the equilateral triangle is equal to two times the base of the right-angle triangles $a = 2\cos(30°) = \sqrt{3}$. To find the area, we use $A_\triangle = \frac{1}{2}ah$, where $h = 1 + \sin(30°)$.

P1.35 We know $\sin^2(\theta) = \frac{1}{2}(1 - \cos(2\theta))$ and $\cos^2(\theta) = \frac{1}{2}(1 + \cos(2\theta))$, so their product is $\frac{1}{4}(1 - \cos(2\theta)\cos(2\theta))$. Note $\cos(2\theta)\cos(2\theta) = \cos^2(2\theta)$. Using the power-reduction formula on the term $\cos^2(2\theta)$ leads to the final answer $\sin^2\theta\cos^2\theta = \frac{1}{4}\left(1 - \frac{1}{2}(1 + \cos(4\theta))\right)$.

P1.36 Split the octagon into eight isosceles triangles. The height of each triangle will be 1, and its angle measure at the centre will be $\frac{360°}{8} = 45°$. Split each of these triangles into two halves down the middle. The octagon is now split into 16 similar right-angle triangles with angle measure 22.5° at the centre. In a right-angle triangle with angle 22.5° and adjacent side 1, what is the length of the opposite side? The opposite side of each of the 16 triangles is $\frac{b}{2} = \tan(22.5°)$, so the perimeter of the octagon is $P_\bigcirc = 16\tan(22.5°)$. In general, if a unit circle is inscribed inside an n-sided regular polygon, the perimeter of the polygon is $P_n = 2n\tan\left(\frac{360°}{2n}\right)$. To find the area of the octagon, we use the formula $A_\triangle = \frac{1}{2}bh$, with $b = 2\tan(22.5°)$ and $h = 1$ to find the area of each isosceles triangle. The area of the octagon is $A_\bigcirc = 8 \cdot \frac{1}{2}(2\tan(22.5°))(1) = 8\tan(22.5°)$. For an n-sided regular polygon the area formula is $A_n = n\tan\left(\frac{360°}{2n}\right)$. Bonus points if you can tell me what happens to the formulas for P_n and A_n as n goes to infinity (see `bit.ly/1jGU1Kz`).

P1.40 Initially the horizontal distance between the observer and the plane is $d_1 = \frac{2000}{\tan 30°}\,\text{m}$. After 10 seconds, the distance is $d_2 = \frac{2000}{\tan 55°}\,\text{m}$. Velocity is change in distance divided by the time $v = \frac{d_1 - d_2}{10} = 206.36\ \text{m/s}$. To convert m/s into km/h, we must multiply by the appropriate conversion factors: $206.36\,\text{m/s} \times \frac{1\,\text{km}}{1000\,\text{m}} \times \frac{3600\,\text{s}}{1\,\text{h}} = 742.92\ \text{km/h}$.

P1.41 The volume of the water stays constant and is equal to $1000\,\text{cm}^3$. Initially the height of the water h_1 can be obtained from the formula for the volume of a cylinder $1000\,\text{cm}^3 = h_1\pi(17\,\text{cm})^2$, so $h_1 = 5.51\,\text{cm}$. After the bottle is inserted, the volume of water has the shape of a cylinder from which a cylindrical part is missing and $1000\,\text{cm}^3 = h_2\left(\pi(17\,\text{cm})^2 - \pi(7.5\,\text{cm})^2\right)$. We find $h_2 = 6.84\,\text{cm}$. The change in water height is $h_2 - h_1 = 1.33\,\text{cm}$.

P1.42 Using the law of cosines for the angles α_1 and α_2, we obtain the equations $7^2 = 8^2 + 12^2 - 2(8)(12)\cos\alpha_1$ and $11^2 = 4^2 + 12^2 - 2(4)(12)\cos\alpha_2$ from which we find $\alpha_1 = 34.09°$ and $\alpha_2 = 66.03°$. In the last step we use the law of cosines again to obtain $x^2 = 8^2 + 4^2 - 2(8)(4)\cos(34.09° + 66.03°)$.

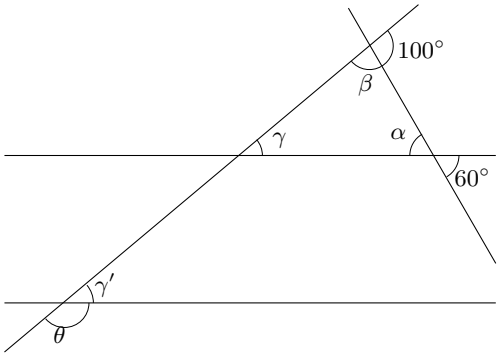
P1.44 The length of the horizontal part of the rope is $\ell_h = 4\sin 40$. The circular portion of the rope that hugs the pulley has length $\frac{1}{4}$ of the circumference of a circle with radius $r = 50\,\text{cm} = 0.5\,\text{m}$. Using the formula $C = 2\pi r$, we find $\ell_c = \frac{1}{4}\pi(0.5)^2 = \frac{\pi}{16}$. The vertical part of the rope has length $\ell_v = 4\cos 40 + 2$. The total length of rope is $\ell_h + \ell_c + \ell_v = 7.83\,\text{m}$.

P1.45 The rectangle's area is equal to its length times its height $A_{\text{rect}} = \ell h$.

P1.46 The box's volume is $V = w \times h \times \ell = 10.5 \times 7 \times 22.3 = 1639\,\text{cm}^3 = 1.639\,\text{L}$.

P1.47 We didn't really cover these concepts in the book, but since we're on the topic let's define some vocabulary. The *complement* of an acute angle is its defect from a right angle; that is, the angle by which it falls short of a right angle. (i) Two angles are complementary when their sum is 90°. The *supplement* of an angle is its defect from two right angles, that is, the angle by which it falls short of 180°. (ii) Two angles are supplementary when their sum is 180°. Angles that are complementary or supplementary to the same angle are equal to one another.

We'll now use these facts and the diagram below to find the angle θ.



The angle α is vertically opposite to the angle 60° so $\alpha = 60°$. The angle β is supplementary to the angle 100° so $\beta = 180 - 100 = 80°$. The sum of the angles in a triangle is 180° so $\gamma = 180° - \alpha - \beta = 40°$. The two horizontal lines are parallel so the diagonally cutting line makes the same angle with them: $\gamma' = \gamma = 40°$. The angle θ is supplementary to the angle γ' so $\theta = 180 - 40 = 120°$.

P1.48 The base of this triangle has length $2r$ and each side has length $R+r$. If you split this triangle through the middle, each half is a right triangle with an angle at the centre $\frac{360°}{24} = 15°$, hypotenuse $R + r$, and opposite side r. We therefore have $\sin 15° = \frac{r}{R+r}$. After rearranging this equation, we find $\frac{R}{r} = \frac{1-\sin 15°}{\sin 15°} = 2.8637$.

P1.51 The tank's total capacity is $15 \times 6 \times 5 = 450\,\text{m}^3$. If 30% of its capacity is spent, then 70% of the capacity remains: $315\,\text{m}^3$. Knowing that $1\,\text{m}^3 = 1000\,\text{L}$, we find there are 315 000 L in the tank.

P1.52 The first tank contains $\frac{1}{4} \times 4000 = 1000\,\text{L}$. The second tank contains three times more water, so 3000 L. The total is 4000 L.

P1.53 Let's define w and h to be the width and the height of the hole. Define d to be the distance from the hole to the sides of the lid. The statement of the problem dictates the following three equations must be satisfied: $w + 2d = 40$, $h + 2d = 30$, and $wh = 500$. After some manipulations, we find $w = 5(1 + \sqrt{21})$, $h = 5(\sqrt{21} - 1)$ and $d = \frac{1}{2}(35 - 5\sqrt{21})$.

P1.54 The amount of wood in a pack of wood is proportional to the area of a circle $A = \pi r^2$. The circumference of this circle is equal to the length of the rope $C = \ell$. Note the circumference is proportional to the radius $C = 2\pi r$. If we want double the area, we need the circle to have radius $\sqrt{2}r$, which means the circumference needs to be $\sqrt{2}$ times larger. If we want a pack with double the wood, we need to use a rope of length $\sqrt{2}\ell$.

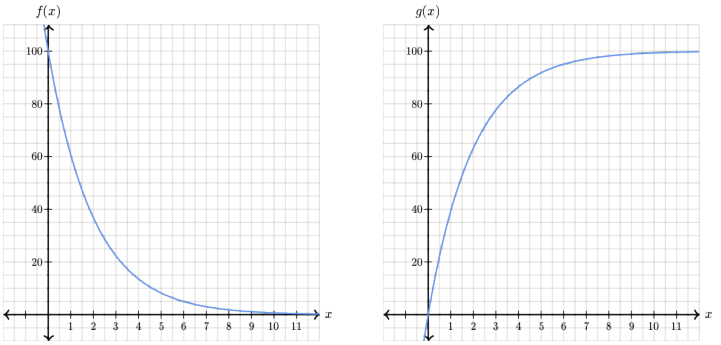
P1.55 In 10 L of a 60% acid solution there are 6 L of acid and 4 L of water. A 20% acid solution will contain four times as much water as it contains acid, so 6 L acid and 24 L water. Since the 10 L we start from already contains 4 L of water, we must add 20 L.

P1.56 The document must have a 768/1004 aspect ratio, so its height must be $6 \times \frac{1004}{768} = 7.84375$ inches.

P1.57 If we rewrite $1+2+3+\cdots+98+99+100$ by pairing numbers, we obtain the sum $(1+100)+(2+99)+(3+98)+\cdots$. This list has 50 terms and each term has the value 101. Therefore $1+2+3+\cdots+100 = 50 \times 101 = 5050$.

P1.62 An nAPR of 12% means the monthly interest rate is $\frac{12\%}{12} = 1\%$. After 10 years you'll owe $\$5000(1.01)^{120} = \$16501.93$. Yikes!

P1.63 The graphs of the functions are shown in Figure A.1. Observe that $f(x)$ decreases to 37% of its initial value when $x = 2$. The increasing exponential $g(x)$ reaches 63% of its maximum value at $x = 2$.



(a) The graph of $f(x)$. (b) The graph of $g(x)$.

Figure A.1: The graphs of the two functions from **P1.63**.

P1.64 We're looking for the time t such that $Q(t)/Q_\text{o} = \frac{1}{2}$, which is the same as $e^{-5t} = 0.5$. Taking logarithms of both sides we find $-5t = \ln(0.5)$, and solving for t we find $t = 0.14\,\text{s}$.

P1.65 We're told $T(24)/T_\text{o} = \frac{1}{2} = e^{-24/\tau}$, which we can rewrite as $\ln(\frac{1}{2}) = -24/\tau$. Solving for τ, we find $\tau = \frac{24}{\ln 2} = 34.625\,\text{min}$. To find the time the body takes to reach 1% of its initial temperature, we must solve for t in $T(t)/T_\text{o} = 0.01 = e^{-t/34.625}$. We find $t = 159.45\,\text{min}$.

P1.67 There exists at least one banker who is not a crook. Another way of saying the same thing is "not all bankers are crooks"—just *most* of them.

P1.68 Everyone steering the ship at Monsanto ought to burn in hell, forever.

P1.69 (a) Investors with money but without connections. (b) Investors with connections but no money. (c) Investors with both money and connections.

Chapter 2 solutions

Answers to problems

P2.1 (a) $\vec{u}_1 = 5\angle 90°$. (b) $\vec{u}_2 = \sqrt{5}\angle 63.4°$. (c) $\vec{u}_3 = \sqrt{5}\angle 243.4°$ or $\sqrt{5}\angle -116.6°$. **P2.2** (a) $\vec{v}_1 = (17.32, 10)$. (b) $\vec{v}_2 = (0, -10)$. (c) $\vec{v}_3 = (-4.33, 2.5)$. **P2.3** (a) $\vec{w}_1 = 9.06\hat{\imath} + 4.23\hat{\jmath}$. (b) $\vec{w}_2 = -7\hat{\jmath}$. (c) $\vec{w}_3 = 3\hat{\imath} - 2\hat{\jmath} + 3\hat{k}$. **P2.4** (a) $(3, 4)$. (b) $(0, 1)$. (c) $(7.33, 6.5)$. **P2.5** $Q = (5.73, 4)$. **P2.6** (1) 6. (2) 0. (3) -3. (4) $(-2, 1, 1)$. (5) $(3, -3, 0)$. (6) $(7, -5, 1)$. **P2.7** $(-\frac{2}{3}, \frac{1}{3}, \frac{2}{3})$ or $(\frac{2}{3}, -\frac{1}{3}, -\frac{2}{3})$. **P2.8** $(12, -4, -12)$. **P2.9** (a) $2i$. (b) $\frac{1}{4}(5 + i)$. (c) $2 + i$. **P2.10** (a) $x = \pm 2i$. (b) $x = -16$. (c) $x = -1 - i$ and $x = -1 + i$. (d) $x = i$, $x = -i$, $x = \sqrt{3}i$, and $x = -\sqrt{3}i$. **P2.11** (a) $\sqrt{5}$. (b) $\frac{1}{2}(-3 + i)$. (c) $-5 - 5i$. **P2.12** $t = 52$ weeks.

Solutions to selected problems

P2.7 See `bit.ly/1cOa8yo` for calculations.

P2.8 Any multiple of the vector $\vec{u}_1 \times \vec{u}_2 = (-3, 1, 3)$ is perpendicular to both $\vec{u}_1$ and $\vec{u}_2$. We must find a multiplier $t \in \mathbb{R}$ such that $t(-3, 1, 3) \cdot (1, 1, 0) = 8$. Computing the dot product we find $-3t + t = 8$, so $t = -4$. The vector we're looking for is $(12, -4, -12)$. See `bit.ly/1nmYH8T` for calculations.

P2.12 We want the final state of the project to be 100% real: $p_f = 100$. Given that we start from $p_i = 100i$, the rotation required is $e^{-i\alpha h(t)} = e^{-i\frac{\pi}{2}}$, which means $\alpha h(t) = \frac{\pi}{2}$. We can rewrite this equation as $h(t) = 0.2t^2 = \frac{\pi}{2\alpha}$ and solving for t we find $t = \sqrt{\frac{\pi}{2(0.002904)(0.2)}} = 52$ weeks.

Chapter 3 solutions

Answers to problems

P3.1 $(1, 2, 3)$. **P3.2** $|a\rangle + |b\rangle = 5|0\rangle + 2|1\rangle$. **P3.3** a) 5; b) $(-1, 1, 1)$; c) $(0, 0, 0)$; d) $(0, 0, 0)$. **P3.4** a) 0; b) $\hat{k} = (0, 0, 1)$; c) $\hat{\jmath} + (-\hat{k}) = (0, 1, -1)$; d) $(-\hat{k}) = (0, 0, -1)$.

Chapter 4 solutions

Answers to exercises

E4.1 $x = 4$, $y = -2$. **E4.3** **(a)** no solution; **(b)** $x = 0$, $y = 2$; **(c)** $\{(1, 1) + s(-1, 1), \forall s \in \mathbb{R}\}$. **E4.4** (1) $X = BA^{-1}$; (2) $X = C^{-1}B^{-1}A^{-1}ED^{-1}$; (3) $X = AD^{-1}$. **E4.5**

$P = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$, $Q = \begin{bmatrix} -5 & 9 \\ 8 & 6 \end{bmatrix}$. **E4.7** $V = 2$. **E4.8** $A^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$. **E4.9** Link to .ods .xls files.

Solutions to selected exercises

E4.2 The row operations required to bring `A` to reduced row echelon form are: $R_1 \leftarrow \frac{1}{3}R_1$, $R_2 \leftarrow R_2 - 2R_1$, $R_2 \leftarrow -2R_2$, $R_1 \leftarrow R_1 - R_2$. Using `SymPy` these operations are implemented as follows:

```
>>> A[0,:] = A[0,:]/3
>>> A[1,:] = A[1,:] - 2*A[0,:]
>>> A[1,:] = -2*A[1,:]
>>> A[0,:] = A[0,:] - A[1,:]
```

Try displaying the matrix `A` after each operation to watch the progress.

E4.7 See `bit.ly/181ugMm` for calculations.

Answers to problems

P4.1 $x = 15$ and $y = 2$. **P4.2** **(a)** $R_2 \leftarrow R_2 - 2R_1$, $R_2 \leftarrow -2R_2$, $R_1 \leftarrow R_1 - R_2$; **(b)** $R_2 \leftarrow R_2 - 2R_1$, $R_2 \leftarrow \frac{-2}{3}R_2$, $R_1 \leftarrow R_1 - \frac{3}{2}R_2$; **(c)** $R_1 \leftarrow \frac{1}{2}R_1$, $R_2 \leftarrow R_2 - 3R_1$, $R_2 \leftarrow \frac{4}{3}R_2$, $R_1 \leftarrow R_1 - \frac{3}{4}R_2$. **P4.3** **(a)** $(-2, 2)$; **(b)** $(-4, -1, -2)$; **(c)** $(\frac{-2}{5}, \frac{-1}{2}, \frac{3}{5})$. **P4.4** **(a)** $\{(2,0)+s(-2,1), \forall s \in \mathbb{R}\}$; **(b)** $\{(2,1,0)+t(3,1,1), \forall t \in \mathbb{R}\}$; **(c)** $\{(\frac{1}{10}, \frac{3}{5}, 0)+\alpha(1,1,0), \forall \alpha \in \mathbb{R}\}$. **P4.5** **(a)** $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} -a_1 \\ -1 \\ -a_2 \end{bmatrix}, \ \forall t \in \mathbb{R} \right\}$. **(b)** $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ 5 \end{bmatrix}$. **P4.6** $C = B^{-1}$. **P4.7** **(a)** $M^{-1}L^{-1}MK$; **(b)** $J^{-3}K^{-1}J^2$; **(c)** $A = \mathbb{1}$; **(d)** $Y = N^{-1}$. **P4.8** $\vec{x} = (-7, -19, -3)^\mathsf{T}$. **P4.9** $\vec{x} = (30.64, 10.48, 17.06)$. **P4.10** a) $AB = \begin{bmatrix} 6 & 2 \\ 10 & 2 \\ 11 & 3 \end{bmatrix}$. b) $AA = \begin{bmatrix} 4 & 6 & 3 \\ 6 & 18 & 8 \\ 3 & 7 & 5 \end{bmatrix}$. c) BA doesn't exist. d) BB doesn't exist. **P4.11** $\begin{bmatrix} -2 & -2 \\ -15 & -15 \end{bmatrix}$. **P4.12** a) $\begin{bmatrix} 0 & \cos(\alpha)-\sin(\alpha) \\ \sin(\alpha)-\cos(\alpha) & -\cos(2\alpha) \end{bmatrix}$; b) $\begin{bmatrix} \cos^2(\alpha) & \sin(\alpha) \\ -\cos(\alpha) & \sin^2(\alpha) \end{bmatrix}$; c) $\begin{bmatrix} \cos(2\alpha) & \sin(\alpha) \\ -\cos(\alpha) & 0 \end{bmatrix}$. **P4.13** a) -3. b) 0. c) 10. **P4.14** $\det(A) = -48$; $\det(B) = 13$. **P4.15** Area $= 2$. **P4.16** Volume $= 8$. **P4.17** a) 86. b) -86. c) -172. **P4.18** For both rows and columns: A: not independent; B: independent; C: not independent; D: not independent. **P4.19** $\det(J) = r$. **P4.20** $|\det(J_s)| = \rho^2 \sin\phi$. **P4.21** a) The inverse doesn't exist. b) $\begin{bmatrix} 5 & -2 \\ -2 & 1 \end{bmatrix}$. c) $\begin{bmatrix} 2 & -\frac{3}{2} \\ -1 & 1 \end{bmatrix}$. **P4.22** $B = \begin{bmatrix} -17 & -30 \\ 5 & 8 \end{bmatrix}$. **P4.23** $A^{-1} = \frac{1}{21}\begin{bmatrix} -3 & -5 & \frac{-21}{6} & 11 \\ 3 & -2 & 7 & -4 \\ 9 & 15 & \frac{21}{2} & -12 \\ 3 & -9 & 0 & 3 \end{bmatrix}$. **P4.25** $C_A = \begin{bmatrix} 1 & 2 & -4 \\ -2 & -1 & 2 \\ 1 & 5 & -7 \end{bmatrix}$; $C_B = \begin{bmatrix} -10 & 6 & -12 \\ -4 & -10 & 20 \\ 1 & 18 & -5 \end{bmatrix}$; $C_C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & -2 & 2 & -2 \\ -4 & 4 & -4 & 4 \\ -8 & 8 & -8 & 8 \end{bmatrix}$. **P4.26** $a = -3$, $b = 1$, $c = 2$, $d = -2$. **P4.27** $\begin{bmatrix} 1 & -1 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -3 \\ 0 & 2 \end{bmatrix}$.

Solutions to selected problems

P4.6 First simplify the equation by multiplying with A^{-1} from the left, and with D^{-1} form the right, to obtain $BC = \mathbb{1}$. Now we can isolate C by multiplying with B^{-1} from the left. We obtain $B^{-1} = C$.

P4.9 Start by rewriting the matrix equations as $(\mathbb{1} - A)\vec{x} = \vec{d}$, then solve for $\vec{x}$ by hitting the equation with the appropriate inverse: $\vec{x} = (\mathbb{1} - A)^{-1}\vec{d}$. See `bit.ly/1hg44Ys` for the details of the calculation.

P4.17 The answers in a) and b) have different signs because interchanging rows in a matrix changes the sign of the determinant. For part c), we use the fact that multiplying one row of a matrix by a constant has the effect of multiplying the determinant by the same constant.

P4.18 To check if the rows/columns of a matrix are independent, we can calculate the determinant of the matrix. If the determinant is not zero then vectors are independent, else vectors are not independent. The columns of a square matrix form a linearly independent set if and only if the rows of the matrix form a linearly independent set.

P4.19 The determinant of J is

$$\begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r\cos^2\theta + r\sin^2\theta = r.$$

P4.20 The determinant of J_s is

$$\begin{aligned} \det(J_s) &= \begin{vmatrix} \sin\phi\cos\theta & -\rho\sin\phi\sin\theta & \rho\cos\phi\cos\theta \\ \sin\phi\sin\theta & \rho\sin\phi\cos\theta & \rho\cos\phi\sin\theta \\ \cos\phi & 0 & -\rho\sin\phi \end{vmatrix} \\ &= \rho^2\left(\cos^2\phi\sin\phi(-1) \;-\; \sin^3\phi(1)\right) \\ &= -\rho^2\sin\phi\left(\cos^2\phi + \sin^2\phi\right) = -\rho^2\sin\phi. \end{aligned}$$

Since we're only interested in finding the *volume factor* we can ignore the sign of the Jacobian's determinant: $|\det(J_s)| = \rho^2\sin\phi$.

P4.22 To solve for the matrix B in the equation $AB = C$, we must get rid of the matrix A on the left side. We do this by multiplying the equation $AB = C$ by the inverse A^{-1}. We find the inverse of A by starting from the array $[\ A \mid \mathbb{1}\ ]$, and performing the row operations $\mathcal{R}_1 : R_2 \leftarrow R_2 - 2R_1$, $\mathcal{R}_2 : R_2 \leftarrow -R_2$, and $\mathcal{R}_3 : R_1 \leftarrow R_1 - 4R_2$, to find the matrix $A^{-1} = \begin{bmatrix} -7 & 4 \\ 2 & -1 \end{bmatrix}$. Applying A^{-1} to both sides of the equation we find $B = A^{-1}C = \begin{bmatrix} -17 & -30 \\ 5 & 8 \end{bmatrix}$.

P4.24 Zero matrix has the $\det(A) = 0$. We have $A^{-1} = \frac{1}{\det(A)}\text{adj}(A)$. We cannot divide by zero, so the zero matrix has no inverse.

P4.26 Multiply two matrices and create systems of four equations: $a + 3c = 3$, $-2a - c = 4$, $b + 3d = -5$, $-2b - d = 0$. Combine the first two equations and the last two to find the variables by using multiplication and addition.

P4.27 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & df+dh \end{bmatrix} = \begin{bmatrix} 2ae & ae+ec \\ ce+ca & c^2+ec \end{bmatrix}$. So we have $2ae = -2$, then $ae = -1$; $ae + ec = -3$, then $ec = -2$; $ec + ca = 0$, then $ca = 2$; $ce + c^2$=2, then $c = 2$. Therefore $c = d = h = 2, a = g = 1$ and $e = b = f = -1$.

Chapter 5 solutions

Answers to exercises

E5.1 $d(\ell, O) = \frac{3\sqrt{291}}{97} \approx 0.5276$. **E5.2** $d(P, O) = 5\sqrt{3} \approx 8.66$. **E5.3** $(-2, 11, 7)\cdot[(x, y, z) - (1, 0, 0)] = 0$. **E5.4** $2x + y = 5$. **E5.5** $(\frac{30}{53}, \frac{5}{53}, \frac{-20}{53})$. **E5.6** $\mathcal{R}(A) = \text{span}\{(1, 3, 0, 0),\ (0, 0, 1, 0),\ (0, 0, 0, 1)\}$, $\mathcal{C}(A) = \text{span}\{(1, 2, 3)^\mathsf{T},\ (3, 7, 9)^\mathsf{T},\ (3, 6, 10)^\mathsf{T}\}$, and $\mathcal{N}(A) = \text{span}\{(-3, 1, 0, 0)^\mathsf{T}\}$. **E5.7** $\mathcal{N}(A) = \text{span}\{(\frac{1}{2},\ 1\ ,\frac{1}{2}\ , 1)\}$.

Solutions to selected exercises

E5.1 Using the formula from page 187, we find

$$\begin{aligned} d(\ell, O) &= \left\| (4,5,6) - \frac{(4,5,6)\cdot(7,8,9)}{7^2+8^2+9^2}(7,8,9) \right\| \\ &= \left\| (4,5,6) - \frac{122}{194}(7,8,9) \right\| \\ &= \left\| \left(\frac{-39}{97}, \frac{-3}{97}, \frac{33}{97} \right) \right\| = \frac{3\sqrt{291}}{97}. \end{aligned}$$

E5.3 We use the reduced row echelon procedure to find the intersection of the two planes. The line of intersection is $\ell_1 : \{(1,0,0) + (1,-3,5)t \,|\, \forall t \in \mathbb{R}\}$, where $(1,0,0)$ is a point on the line of intersection and $\vec{v}_1 = (1,-3,5)$ is its direction vector. We want to find the equation of a plane $\vec{n} \cdot [(x,y,z) - p_o] = 0$ whose normal vector is perpendicular to both $\vec{v}_1 = (1,-3,5)$ and $\vec{v}_2 = (2,1,-1)$. To find a normal vector use $\vec{n} = \vec{v}_1 \times \vec{v}_2 = (-2,11,7)$. The point $p_o = (1,0,0)$ is on the line ℓ_1, so the equation of the plane is $(-2,11,7) \cdot [(x,y,z) - (1,0,0)] = 0$.

E5.5 The vectors $(2,-4,2)$ and $(6,1,-4)$ define a plane P. We're looking for the projection of $\vec{v}$ on P. First we use the cross product to find a normal vector to the plane P: $\vec{n} = (2,-4,2) \times (6,1,-4) = (14,20,26)$. Then we compute the projection using the formula $\Pi_P(\vec{v}) = \vec{v} - \Pi_{\vec{n}}(\vec{v}) = (\frac{30}{53}, \frac{5}{53}, \frac{-20}{53})$.

E5.7 The null space of A consists of all vectors $\vec{x} = (x_1, x_2, x_3, x_4)^\mathsf{T}$ that satisfy $A\vec{x} = \vec{0}$. To solve this equation we first compute the RREF of A:

$$\text{rref}(A) = \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -\frac{1}{2} \end{bmatrix},$$

and observe it pivots in the first three columns, while $x_4 = s$ is a free variable. The null space of A is

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} 1x_1 - \tfrac{1}{2}s &= 0 \\ 1x_2 - s &= 0 \\ x_3 - \tfrac{1}{2}s &= 0 \end{aligned}$$

The null space is $\mathcal{N}(A) = \text{span}\{(\frac{1}{2}, 1, \frac{1}{2}, 1)\}$.

Answers to problems

P5.1 a) $q = (1,2)$; b) infinite intersection points; c) $m = (1,1)$. **P5.2** a) ℓ_1: $1(x) - 1(y) + 1(z) = 0$; b) ℓ_2: $0(x) + 1(y) + 1(z) = 0$. **P5.3** a) parallel; b) neither; c) perpendicular. **P5.4** $d(q,P) = 1$. **P5.5** a) $d(p,q) = 6$; b) $d(m,n) = 5$; c) $d(r,s) = 3$; d) $d(p,j) = \sqrt{19}$. **P5.6** $x + y + 2z = 4$. **P5.7** $\ell : \left\{ \frac{x+3}{2} = \frac{y-1}{-4} = \frac{z}{-1} \right\}$. **P5.8** $\Pi_{\vec{u}}(\vec{v}) = \frac{\vec{v}\cdot\vec{u}}{\|\vec{u}\|^2}\vec{u} = \frac{(1,1,1)\cdot(2,1,-1)}{2^2+1^2+(-1)^2}(2,1,-1) = \frac{1}{3}(2,1,-1)$; $\Pi_{\vec{v}}(\vec{u}) = \frac{\vec{v}\cdot\vec{u}}{\|\vec{v}\|^2}\vec{v} = \frac{(1,1,1)\cdot(2,1,-1)}{1^2+1^2+1^2}(1,1,1) = \frac{2}{3}(1,1,1)$. **P5.9** $\Pi_P(\vec{v}) = \frac{1}{7}(17,30,-1)$. **P5.10** $\Pi_{P^\perp}(\vec{u}) = \frac{-1}{41}(6,-5,12)$. **P5.11** $d(\ell,P) = \frac{7}{3}$. **P5.16** $A = \begin{bmatrix} 1 & 2x \\ 0 & 1 \end{bmatrix}$. This matrix A corresponds to a *shear* on the x-axis.

Solutions to selected problems

P5.1 To find an intersection point, you have to isolate a variable in one equation and replace it in the second equation, and then find a variable.

P5.3 a) Find a normal for each plane $n_1 = (1,-1,-1)$ and $n_2 = (2,-2,-2) = 2(1,-1,-1)$. These places are multiples of each other so the planes are parallel. b) $n_1 = (3,2,0), n_2 = (0,1,-1)$. $n_1 \cdot n_2 = 2$. Planes are neither parallel or perpendicular. c) $n_1 = (1,-2,1), n_2 = (1,1,1)$. $n_1 \cdot n_2 = 0$. Therefore two planes are perpendicular.

P5.4 $d(q,P) = \frac{|2(2)+1(3)-2(5)|}{\sqrt{2^2+1^2+(-2)^2}} = \frac{3}{3} = 1$.

P5.6 First we find vectors in the place, for example $\vec{u} \equiv r - q = (-1,-1,1)$ and $\vec{v} \equiv s - q = (0,-2,1)$. Then we need to find the normal vector $\vec{n}$, $\vec{n} = \vec{u} \times \vec{v} = (-1,-1,1) \times (0,-2,1) = (1,1,2)$. We can use any of the three points as the point p_o in the geometric equation $\vec{n} \cdot [(x,y,z) - p_o] = 0$. Using $q = (1,3,0)$, we obtain the equation $(1,1,2)\cdot[(x,y,z)-(1,3,0)] = 1(x-1)+1(y-3)+2z = 0$. Simplifying the expression gives $x - 1 + y - 3 + 2z = 0$ and we end up with $x + y + 2z = 4$.

P5.9 Let's first find $\Pi_{P^\perp}(\vec{v}) = \frac{\vec{v}\cdot\vec{n}}{\|\vec{n}\|^2}\vec{n} = \frac{(3,4,1)\cdot(2,-1,4)}{2^2+(-1)^2+4^2}(2,-1,4) = \frac{2}{7}(2,-1,4)$. Then $\Pi_P(\vec{v}) = \vec{v} - \Pi_{P^\perp}(\vec{v}) = (3,4,1) - \frac{2}{7}(2,-1,4) = \frac{1}{7}(17,30,-1)$. We can verify $\Pi_P(\vec{v}) + \Pi_{P^\perp}(\vec{v}) = \frac{1}{7}(17,30,-1) + \frac{2}{7}(2,-1,4) = (3,4,1) = \vec{v}$. This shows that the projection we found is correct.

P5.10 First we find the normal $\vec{n}$ of the plane P using the cross-product trick $\vec{n} = (s-m) \times (r-m)$. Since $s - m = (-1,-6,3)$ and $r - m = (-2,0,-1)$, we find $\vec{n} = (-1,-6,3) \times (-2,0,-1) = (6,-5,12)$. Now we want to find the projection of $\vec{u}$ onto the space perpendicular to P, which is $\Pi_{P^\perp}(\vec{u}) = \frac{\vec{v}\cdot\vec{n}}{\|\vec{n}\|^2}\vec{n} = \frac{(-2,1,1)\cdot(6,-5,12)}{6^2+(-5)^2+12^2}(6,-5,12) = \frac{-1}{41}(6,-5,12)$.

P5.11 We'll compute the distance by finding a vector $\vec{v}$ that connects an arbitrary point on the plane P with an arbitrary point on the line ℓ and then computing the component of $\vec{v}$ that is perpendicular to the plane. The point that lies on the line is $p_\ell = (1,-3,2)$ and the point on the plane is $q_P = (0,1,1)$. The vector between them is $\vec{v} = (0,1,1) - (1,-3,2) = (1,-4,1)$. To compute $d(\ell,P)$ we must find $\frac{|\vec{n}\cdot\vec{v}|}{\|\vec{n}\|} = \frac{(1,-4,1)\cdot(-1,2,2)}{\sqrt{(-1)^2+2^2+2^2}} = \frac{7}{3}$.

P5.12 Directly check that the sum of two upper triangular matrices and scalar multiplication result in upper triangular matrices. And of course the zero matrix is upper triangular.

P5.13 If A is a diagonal matrix, we have $A_{ij} = 0 = A_{ji}$ when $i \neq j$. Therefore diagonal matrices are symmetric.

P5.14 If $\{\vec{u},\vec{v}\}$ is a basis then the dimension of V would be two. So it is enough to check both $\{\vec{u}+\vec{v}, a\vec{u}\}$ and $\{a\vec{u}, b\vec{v}\}$ are linearly independent. Assuming $s(\vec{u}+\vec{v}) + ta\vec{u} = (s+ta)\vec{u} + s\vec{v} = 0$ we have $s + ta = s = 0$ and hence $s = t = 0$. Assuming $sa\vec{u} + tb\vec{v} = 0$ we have $sa = tb = 0$ and hence $s = t = 0$.

P5.15 Let us assume that $\{\vec{v}_3, \vec{v}_2+\vec{v}_3, \vec{v}_1+\vec{v}_2+\vec{v}_3\}$ is not linearly independent. Then $x_1\vec{v}_3 + x_2(\vec{v}_2+\vec{v}_3) + x_3(\vec{v}_1+\vec{v}_2+\vec{v}_3) = 0$ has a nontrivial solution. Let us say this solution is (c_1,c_2,c_3) where $c_1 \neq 0$ or $c_2 \neq 0$ or $c_3 \neq 0$. Then we see that

$$\begin{aligned} 0 &= c_1\vec{v}_3 + c_2(\vec{v}_2+\vec{v}_3) + c_3(\vec{v}_1+\vec{v}_2+\vec{v}_3) \\ &= c_1\vec{v}_3 + c_2\vec{v}_2 + c_2\vec{v}_3 + c_3\vec{v}_1 + c_3\vec{v}_2 + c_3\vec{v}_3 \\ &= (c_1+c_2+c_3)\vec{v}_3 + (c_2+c_3)\vec{v}_2 + c_3\vec{v}_1. \end{aligned}$$

If $c_1 \neq 0$, then $c_1+c_2+c_3 \neq 0$. If $c_2 \neq 0$, then $c_2+c_3 \neq 0$. If $c_3 \neq 0$, then $c_3 \neq 0$. This means that $x_1\vec{v}_1 + x_2\vec{v}_2 + x_3\vec{v}_3 = 0$ has a nontrivial solution. In other words, we deduce that $\{\vec{v}_1,\vec{v}_2,\vec{v}_3\}$ is linearly dependent. It is a contradiction. Because we are given that $\{\vec{v}_1,\vec{v}_2,\vec{v}_3\}$ is linearly independent. This contradiction is coming from the assumption that $\{\vec{v}_3, \vec{v}_2+\vec{v}_3, \vec{v}_1+\vec{v}_2+\vec{v}_3\}$ is not linearly independent. Therefore, $\{\vec{v}_3, \vec{v}_2+\vec{v}_3, \vec{v}_1+\vec{v}_2+\vec{v}_3\}$ must be linearly independent.

Chapter 6 solutions

Answers to exercises

E6.1 $\text{Dom}(T) = \mathbb{R}^3$, $\text{Co}(T) = \mathbb{R}^3$, $\text{Ker}(T) = \{\vec{0}\}$, $\text{Im}(T) = \mathbb{R}^3$. T is injective and surjective, and therefore bijective. **E6.2** ${}_B[M_T]_B = {}_B[\mathbb{1}]_{B'}\,{}_{B'}[M_T]_{B'}\,{}_{B'}[\mathbb{1}]_B$.

Solutions to selected exercises

E6.2 Start from the formula for ${}_{B'}[M_T]_{B'}$ and multiply it by ${}_B[\mathbb{1}]_{B'}$ from the left and by ${}_{B'}[\mathbb{1}]_B$ from the right:

$$\begin{aligned} {}_B[\mathbb{1}]_{B'}\,{}_{B'}[M_T]_{B'}\,{}_{B'}[\mathbb{1}]_B &= {}_B[\mathbb{1}]_{B'}\,{}_{B'}[\mathbb{1}]_B\,{}_B[M_T]_B\,{}_B[\mathbb{1}]_{B'}\,{}_{B'}[\mathbb{1}]_B \\ &= {}_B[M_T]_B\,. \end{aligned}$$

Recall that ${}_B[\mathbb{1}]_{B'}$ is the inverse matrix of ${}_{B'}[\mathbb{1}]_B$ so the two matrices cancel out.

E6.3 Let $\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_n$ be the n columns of A. Since the columns of A form a basis for $\mathbb{R}^n$, any vector $\vec{b} \in \mathbb{R}^n$ can be written as a unique linear combination of the columns of A: $\vec{b} = x_1\vec{c}_1 + x_2\vec{c}_2 + \cdots + x_n\vec{c}_n$ for some coefficients $x_1, x_2, \ldots, x_n$. Reinterpreting the last equation as a matrix product in the column picture, we conclude that $A\vec{x} = \vec{b}$ has a unique solution $\vec{x}$.

E6.4 Let $B_s = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$ be the standard basis for $\mathbb{R}^n$. Since $A\vec{x} = \vec{b}$ has a solution $\vec{x}$ for every possible $\vec{b}$, it is possible to find the solutions $\vec{x}_i$ in n equations of the form $A\vec{x}_i = \hat{e}_i$, for $i \in \{1, 2, \ldots, n\}$. Now construct the matrix B that contains the solutions $\vec{x}_i$ as columns: $B = [\vec{x}_1, \ldots, \vec{x}_n]$. Observe that $AB = A[\vec{x}_1, \ldots, \vec{x}_n] = [A\vec{x}_1, \ldots, A\vec{x}_n] = [\hat{e}_1, \ldots, \hat{e}_n] = \mathbb{1}_n$. The equation $BA = \mathbb{1}_n$ implies $B \equiv A^{-1}$ and thus A is invertible.

Answers to problems

P6.1 $\text{Im}(T) = \text{span}\{(1,1,0), (0,-1,2)\}$. **P6.2** $\left[\begin{smallmatrix} 3 & 1 \\ -1 & 3 \end{smallmatrix}\right]$. **P6.3** $D = \left[\begin{smallmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{smallmatrix}\right]$.

Solutions to selected problems

P6.1 Applying T to the input vector $(1,0)$ produces $(1, 1-0, 2\cdot 0) = (1,1,0)$, and the input vector $(0,1)$ produces the output $(0, 0-1, 2\cdot 1) = (0,-1,2)$. Thus, $\text{Im}(T) = \text{span}\{(1,1,0), (0,-1,2)\} \subseteq \mathbb{R}^3$.

P6.2 We apply L to both $e^{2x}\cos x$ and $e^{2x}\sin x$

$$L(e^{2x}\cos x) = 2e^{2x}\cos x - e^{2x}\sin x + e^{2x}\cos x = 3e^{2x}\cos x - e^{2x}\sin x$$

$$L(e^{2x}\sin x) = 2e^{2x}\sin x + e^{2x}\cos x + e^{2x}\sin x = e^{2x}\cos x + 3e^{2x}\sin x.$$

The first corresponds to the vector $[3,-1]$,with respect to the given basis, and the second corresponds to the vector $[1,3]$ with respect to the given basis. These vectors are columns of the matrix representing L. Thus, the matrix representing L with respect to the given basis is $\left[\begin{smallmatrix} 3 & 1 \\ -1 & 3 \end{smallmatrix}\right]$.

Chapter 7 solutions

Answers to exercises

E7.3 See `math.stackexchange.com/a/29374/46349`. **E7.6** $\alpha = -3$ and $\beta = 4$. **E7.7** $(1,1)^\mathsf{T}$ and $(1,-2)^\mathsf{T}$ are eigenvectors of L. **E7.8** No. **E7.9** $\det(A) = 15$,

$A^{-1} = \begin{bmatrix} 1 & -\frac{4}{5} & -\frac{11}{3} \\ 0 & \frac{1}{5} & -\frac{1}{3} \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$; $\det(B) = xz$, $B^{-1} = \begin{bmatrix} \frac{1}{x} & 0 \\ -\frac{y}{xz} & \frac{1}{z} \end{bmatrix}$; $\det(C) = 1$, $C^{-1} = \begin{bmatrix} 5 & 0 \\ 0 & \frac{1}{5} \end{bmatrix}$. **E7.10** $\vec{v} = (1, 0, 0, 1)$. **E7.11** $V = [1, 0, 0, 1]$. **E7.12** The solution space is one-dimensional and spanned by $f(t) = e^{-t}$. **E7.13** Yes. **E7.14** $\vec{e}_1 = (4, 2)$, $\vec{e}_2 = (-1, 2)$. **E7.15** $\hat{e}_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$, $\hat{e}_2 = (\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}})$, and $\hat{e}_3 = (-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. **E7.16** $\hat{\mathbf{e}}_1 = \frac{1}{\sqrt{2}}$, $\hat{\mathbf{e}}_2 = \sqrt{3/2}x$, and $\hat{\mathbf{e}}_3 = \frac{\sqrt{5}}{\sqrt{8}}(3x^2 - 1)$.

E7.17 $Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$, and $R = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{bmatrix}$. **E7.18** (a) $-1 + i$; (b) $-14 + 23i$; (c) $\frac{1}{25}(26 + 7i)$.

Solutions to selected exercises

E7.1 The characteristic polynomial of the matrix has degree n, and an n^{th}-degree polynomial has at most n distinct roots.

E7.3 Proof by contradiction. Assume you have n *distinct* eigenvalues λ_i and eigenvectors $\{\vec{e}_i\}$ which are linearly dependent: $\sum_{i=1}^{n} \alpha_i \vec{e}_i = \vec{0}$ with some $\alpha_i \neq 0$. If a nonzero combination of α_i could give the zero vector as a linear combination, the equation $(A - \lambda_n I)(\sum \alpha_i \vec{e}_i) = (A - \lambda_n I)\vec{0} = \vec{0}$ would be true. However, if you expand the expression on the left, you'll see it's not equal to zero.

E7.4 Multiply both sides of the eigenvalue equation $A\vec{e}_\lambda = \lambda\vec{e}_\lambda$ by A to obtain $AA\vec{e}_\lambda = \lambda A\vec{e}_\lambda = \lambda^2\vec{e}_\lambda$. Thus λ^2 is an eigenvalue of A^2.

E7.5 Multiply both sides of the eigenvalue equation $A\vec{e}_\lambda = \lambda\vec{e}_\lambda$ by A^{-1} to obtain $A^{-1}A\vec{e}_\lambda = \lambda A^{-1}\vec{e}_\lambda$. After A^{-1} cancels with A, we're left with the equation $\vec{e}_\lambda = \lambda A^{-1}\vec{e}_\lambda$. Dividing both sides of the equation by λ we obtain $\frac{1}{\lambda}\vec{e}_\lambda = A^{-1}\vec{e}_\lambda$, which shows that λ^{-1} is an eigenvalue of A^{-1}.

E7.6 The characteristic polynomial of A is $p_A(\lambda) = x^2 - \beta x - \alpha$. If we want the eigenvalues of A to be 1 and 3, we must choose α and β so that $p_A(\lambda) = (\lambda - 1)(\lambda - 3)$. Expanding the factored expression, we find $(\lambda - 1)(\lambda - 3) = \lambda^2 - 4\lambda + 3$, so $\alpha = -3$ and $\beta = 4$.

E7.7 First we compute $L(1, 1)^\mathsf{T} = (5, 5)^\mathsf{T} = 5(1, 1)^\mathsf{T}$ so $(1, 1)^\mathsf{T}$ is an eigenvector, with eigenvalue $\lambda = 5$. Next we compute $L(1, -1)^\mathsf{T} = (1, 3)^\mathsf{T} \neq \alpha(1, -1)^\mathsf{T}$ so $(1, -1)^\mathsf{T}$ is not an eigenvector. We can also compute $L(1, -2)^\mathsf{T} = (-1, 2)^\mathsf{T} = -1(1, -2)^\mathsf{T}$, which implies $(1, -2)^\mathsf{T}$ is an eigenvector with eigenvalue -1. Since 2×2 matrix can have at most two eigenvectors, we don't need to check $(2, -1)^\mathsf{T}$—we know its not an eigenvector.

E7.8 A matrix A is Hermitian if it satisfies $A^\dagger = A$, which is not the case for the given matrix.

E7.12 The solutions to the differential equation $f'(t) + f(t) = 0$ are of the form $f(t) = Ce^{-t}$, where C is an arbitrary constant. Since any solution in the *solution space* can be written as a multiple of the function e^{-t}, we say the solution space is *spanned* by e^{-t}. Since one function is sufficient to span the solution space, the solution space is one-dimensional.

E7.13 Consider an arbitrary second-degree polynomial $p(x) = a_0 + a_1 x + a_2 x^2$. We can rewrite it as

$$\begin{aligned} p(x) &= a_0 + a_1[(x - 1) + 1] + a_2[(x - 1) + 1]^2 \\ &= a_0 + a_1(x - 1) + a_1 + a_2[(x - 1)^2 + 2(x - 1) + 1] \\ &= (a_0 + a_1 + a_2)1 + (a_1 + 2a_2)(x - 1) + a_2(x - 1)^2. \end{aligned}$$

Since we've expressed an arbitrary polynomial of degree 2 in the desired form, the answer is yes. In other words, the set $\{1, x-1, (x-1)^2\}$ is a basis for the vector space of polynomials of degree at most 2.

E7.14 The exercise does not require us to normalize the vectors, so we can leave the first vector as is $\vec{v}_1 = \vec{e}_1 = (4,2)$. Next, we calculate $\vec{e}_2$ using the formula $\vec{e}_2 = \vec{v}_2 - \Pi_{\vec{e}_1}(\vec{v}_2)$, which corresponds to removing the component of $\vec{v}_2$ that lies in the direction of $\vec{e}_1$. Using the projection formula we obtain $\Pi_{\vec{e}_1}(\vec{v}_2) \equiv \frac{(4,2)\cdot(1,3)}{\|(4,2)\|^2}(4,2) = \frac{4+6}{16+4}(4,2) = \frac{1}{2}(4,2) = (2,1)$. Thus $\vec{e}_2 = \vec{v}_2 - \Pi_{\vec{e}_1}(\vec{v}_2) = (1,3) - (2,1) = (-1,2)$. Verify that $\vec{e}_1 \cdot \vec{e}_2 = 0$.

Answers to problems

P7.1 **(a)** $\lambda_1 = 6$, $\lambda_2 = -1$; **(b)** $\lambda_1 = -2$, $\lambda_2 = 2$, $\lambda_3 = 0$. **P7.2** $\lambda_1 = \varphi \equiv \frac{1+\sqrt{5}}{2} = 1.6180339\ldots$; $\lambda_2 = -\frac{1}{\varphi} = \frac{1-\sqrt{5}}{2} = -0.6180339\ldots$. **P7.3** $\lambda_1 = \varphi$ and $\lambda_2 = -\frac{1}{\varphi}$. **P7.4** $X = Q^{-1} = \begin{bmatrix} \frac{5+\sqrt{5}}{10} & \frac{\sqrt{5}}{5} \\ \frac{5-\sqrt{5}}{10} & -\frac{\sqrt{5}}{5} \end{bmatrix}$. **P7.5** **(a)** $\lambda_1 = 5$, $\lambda_2 = 4$; **(b)** $\lambda_1 = \frac{1}{2}(5+\sqrt{(5)})$, $\lambda_2 = \frac{1}{2}(5-\sqrt{(5)})$; **(c)** $\lambda_1 = 3, \lambda_2 = \lambda_3 = 0$ and **(d)** $\lambda_1 = -3, \lambda_2 = -1, \lambda_3 = 1$. **P7.6** **(a)** $\lambda_1 = 1$, $\vec{e}_{\lambda_1} = (1,1)^\mathsf{T}$, $\lambda_2 = -1$, $\vec{e}_{\lambda_2} = (1,-1)^\mathsf{T}$; **(b)** $\lambda_1 = 3$, $\vec{e}_{\lambda_1} = (1,3,9)^\mathsf{T}$, $\lambda_2 = 2$, $\vec{e}_{\lambda_2} = (1,2,4)^\mathsf{T}$, $\lambda_3 = -1$, $\vec{e}_{\lambda_3} = (1,-1,1)^\mathsf{T}$. **P7.7** $A^{10} = \begin{bmatrix} 2 & 2 \\ 5 & -1 \end{bmatrix}^{10} = \begin{bmatrix} 765854 & 282722 \\ 706805 & 341771 \end{bmatrix}$. **P7.8** $(x_\infty, y_\infty, z_\infty)^\mathsf{T} = \frac{1}{6}(x_0 + 4y_0 + z_0, x_0 + 4y_0 + z_0, x_0 + 4y_0 + z_0)^\mathsf{T}$. **P7.9** **(a)** Yes; **(b)** No; **(c)** Yes. **P7.10** No. **P7.11** No. **P7.14** $\hat{e}_1 = (0,1)$, $\hat{e}_2 = (-1,0)$. **P7.15** $\hat{e}_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, $\hat{e}_2 = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. **P7.16** $\hat{e}_1 = (\frac{3}{\sqrt{10}}, \frac{1}{\sqrt{10}})$, $\hat{e}_2 = (\frac{-1}{\sqrt{10}}, \frac{3}{\sqrt{10}})$. **P7.17** $A = Q\Lambda Q^{-1} = \begin{bmatrix} 2 & 0 & -5 \\ 0 & 2 & 0 \\ 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$. **P7.18** a) 5;b) $2+3i$; c) $2+5i$; d) $\sqrt{26}$. **P7.19** $A + B = \begin{bmatrix} 4 & 2 \\ 8+3i & -5+3i \end{bmatrix}$; $CB = \begin{bmatrix} 3+8i & 2-i \\ 45+3i & -33+31i \\ 16-4i & 16+8i \end{bmatrix}$; $(2+i)B = \begin{bmatrix} 5 & 8-i \\ 9+7i & -15+5i \end{bmatrix}$. **P7.20** **(a)** $\lambda_1 = 2+i$ and $\lambda_2 = 2-i$; **(b)** $\lambda_1 = 3\sqrt{3}i$ and $\lambda_2 = 3\sqrt{3}i$; **(c)** $\lambda_1 = 2+8i$ and $\lambda_2 = 2-8i$. **P7.21** $\left\{\mathbf{e}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{e}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right\}$. **P7.22** $d = 6$. **P7.23** $d = 9$. **P7.24** **(a)** Yes. **(b)** No. **(c)** Yes. **(d)** No. **(e)** Yes. **(f)** Yes. **P7.25** $Q = \begin{bmatrix} \frac{-1+i}{2} & \frac{1+i}{2} \end{bmatrix}$, $\Lambda = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$. **P7.26** **(a)** 9; **(b)** 3; **(c)** $\sqrt{10}$.

Solutions to selected problems

P7.2 To find the eigenvalues of the matrix A we must find the roots of it's characteristic polynomial

$$p(\lambda) = \det(A - \lambda\mathbb{1}) = \det\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = \det\left(\begin{bmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{bmatrix}\right).$$

Using the determinant formula $\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$, we find the characteristic polynomial of A is $p(\lambda) = \lambda^2 - \lambda - 1$. The eigenvalues of A are the roots λ_1 and λ_2 of this equation, which we can find using the formula for solving quadratic equations we saw in Section 1.6 (see page 28).

P7.3 The vector $\vec{e}_1$ is an eigenvector of A because $A\vec{e}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ \frac{1}{\varphi} \end{bmatrix} = \begin{bmatrix} 1+\frac{1}{\varphi} \\ 1 \end{bmatrix}$. Now observe the following interesting fact: $\frac{1}{\varphi}(1 + \frac{1}{\varphi}) = \frac{1}{\varphi} + \frac{1}{\varphi^2} = \frac{\varphi+1}{\varphi^2} = 1$. This means we can write $A\vec{e}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ \frac{1}{\varphi} \end{bmatrix} = \varphi\begin{bmatrix} 1 \\ \frac{1}{\varphi} \end{bmatrix}$, which shows that $\vec{e}_1$ is an eigenvector of A and it corresponds to eigenvalue $\lambda_1 = \varphi$. Similar reasoning

shows $A\vec{e}_2 = -\frac{1}{\varphi}\vec{e}_2$ so $\vec{e}_2$ is an eigenvector of A that corresponds to eigenvalue $\lambda_2 = -\frac{1}{\varphi}$.

P7.4 The eigendecomposition of matrix A is $A = Q\Lambda Q^{-1}$. The unknown matrix X is the inverse matrix of the matrix $Q = \begin{bmatrix} 1 & 1 \\ \frac{1}{\varphi} & -\varphi \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \frac{2}{1+\sqrt{5}} & -\frac{1+\sqrt{5}}{2} \end{bmatrix}$. To find Q^{-1} we can start from the array $[\, Q \mid \mathbb{1} \,]$ and perform row operations until we obtain $[\, \mathbb{1} \mid Q^{-1} \,]$.

P7.6 (a) First we obtain the characteristic polynomial.
(b) The characteristic polynomial is

P7.7 First we decompose A as the product of three matrices $A = Q\Lambda Q^{-1}$, where Q is a matrix of eigenvectors, and Λ contains the eigenvalues of A. $A = \left[\begin{smallmatrix} 2 & 2 \\ 5 & -1 \end{smallmatrix}\right] = \left[\begin{smallmatrix} -2 & 1 \\ 5 & 1 \end{smallmatrix}\right]\left[\begin{smallmatrix} -3 & 0 \\ 0 & 4 \end{smallmatrix}\right]\begin{bmatrix} \frac{-1}{7} & \frac{1}{7} \\ \frac{5}{7} & \frac{2}{7} \end{bmatrix}$. Since the matrix Λ is diagonal, we can compute it's fifth power. $\Lambda^{10} = \begin{bmatrix} 59049 & 0 \\ 0 & 1048576 \end{bmatrix}$ Thus expressing the calculation of A^{10}.

$$A^{10} = \begin{bmatrix} -2 & 1 \\ 5 & 1 \end{bmatrix}\begin{bmatrix} 59049 & 0 \\ 0 & 1048576 \end{bmatrix}\begin{bmatrix} \frac{-1}{7} & \frac{1}{7} \\ \frac{5}{7} & \frac{2}{7} \end{bmatrix} = \begin{bmatrix} 765854 & 282722 \\ 706805 & 341771 \end{bmatrix}.$$

P7.8 The eigenvalues of M are $\frac{1}{4}$, $\frac{1}{2}$, and 1, and its eigendecomposition is $M = Q\Lambda Q^{-1}$. We can compute $(x_\infty, y_\infty, z_\infty)^\mathsf{T}$ using $M^\infty(x_0, y_0, z_0)^\mathsf{T}$. To compute M^∞, we can compute Λ^∞. The $\frac{1}{4}$ and $\frac{1}{2}$ eigenspaces will disappear, so we'll be left only with the subspace of the eigenvalue 1. $M^\infty = Q\Lambda^\infty Q^{-1}$, and each row of this matrix has the form $[\frac{1}{6}, \frac{4}{6}, \frac{1}{6}]$. See `bit.ly/eigenex001` for details.

P7.9 To check if the matrix is orthogonal the transpose of that matrix should act as an inverse such that $O^T O = \mathbb{1}$.

P7.10 A vector space would obey $0 \cdot (a_1, a_2) = (0, 0)$ (the zero vector), but we have $0 \cdot (a_1, a_2) = (0, a_2) \neq (0, 0)$, so $(V, \mathbb{R}, +, \cdot)$ is not a vector space.

P7.11 The vector addition operation is not associative: we have $((a_1, a_2) + (b_1, b_2)) + (c_1, c_2) = (a_1 + 2b_1 + 2c_1, a_2 + 3b_2 + 3c_2)$ but $(a_1, a_2) + ((b_1, b_2) + (c_1, c_2)) = (a_1 + 2b_1 + 4c_1, a_2 + 3b_2 + 9c_2)$.

P7.12 If $\mathbf{v} = \mathbf{0}$, then the inequality holds trivially. If $\mathbf{v} \neq 0$, we can start from the following which holds for any $c \in \mathbb{C}$:

$$\begin{aligned} 0 &\leq \langle \mathbf{u} - c\mathbf{v}, \mathbf{u} - c\mathbf{v} \rangle \\ &= \langle \mathbf{u}, \mathbf{u} - c\mathbf{v} \rangle - c\langle \mathbf{v}, \mathbf{u} - c\mathbf{v} \rangle \\ &= \langle \mathbf{u}, \mathbf{u} \rangle - \overline{c}\langle \mathbf{u}, \mathbf{v} \rangle - c\langle \mathbf{v}, \mathbf{u} \rangle + |c|^2 \langle \mathbf{v}, \mathbf{v} \rangle. \end{aligned}$$

This is true in particular when $c = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle}$, so we continue

$$\begin{aligned} 0 &\leq \langle \mathbf{u}, \mathbf{u} \rangle - \frac{\overline{\langle \mathbf{u}, \mathbf{v} \rangle}\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} - \frac{\langle \mathbf{u}, \mathbf{v} \rangle\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} + \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^2 \langle \mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle^2} \\ 0 &\leq \langle \mathbf{u}, \mathbf{u} \rangle - \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^2}{\langle \mathbf{v}, \mathbf{v} \rangle} - \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^2}{\langle \mathbf{v}, \mathbf{v} \rangle} + \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^2}{\langle \mathbf{v}, \mathbf{v} \rangle} \\ 0 &\leq \langle \mathbf{u}, \mathbf{u} \rangle - \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^2}{\langle \mathbf{v}, \mathbf{v} \rangle} \\ 0 &\leq \langle \mathbf{u}, \mathbf{u} \rangle \langle \mathbf{v}, \mathbf{v} \rangle - |\langle \mathbf{u}, \mathbf{v} \rangle|^2, \end{aligned}$$

from which we conclude $|\langle \mathbf{u}, \mathbf{v} \rangle|^2 \leq \|\mathbf{u}\|^2 \|\mathbf{v}\|^2$. Taking the square root on both sides we obtain the the Cauchy–Schwarz inequality $|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\| \|\mathbf{v}\|$.

P7.13 We proceed using the following chain of inequalities:

$$\begin{aligned}\|\mathbf{u}+\mathbf{v}\|^2 &= \langle \mathbf{u}+\mathbf{v}, \mathbf{u}+\mathbf{v}\rangle \\ &= \langle \mathbf{u},\mathbf{u}\rangle + \langle \mathbf{u},\mathbf{v}\rangle + \langle \mathbf{v},\mathbf{u}\rangle + \langle \mathbf{v},\mathbf{v}\rangle \\ &\leq \|\mathbf{u}\|^2 + 2\langle \mathbf{u},\mathbf{v}\rangle + \|\mathbf{v}\|^2 \\ &\leq \|\mathbf{u}\|^2 + 2\|\mathbf{u}\|\|\mathbf{v}\| + \|\mathbf{v}\|^2 \\ &= (\|\mathbf{u}\| + \|\mathbf{v}\|)^2\end{aligned}$$

Therefore we obtain the equation $\|\mathbf{u}+\mathbf{v}\|^2 \leq (\|\mathbf{u}\| + \|\mathbf{v}\|)^2$, and since $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ are non-negative numbers, we can take the square root on both sides of the equation to obtain $\|\mathbf{u}+\mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$.

P7.14 Let $\vec{v}_1 = (0,1)$ and $\vec{v}_2 = (-1,0)$. We don't really need to perform the Gram–Shmidt procedure since $\vec{v}_2$ is already perpendicular to $\vec{v}_1$ and both vectors have unit length.

P7.15 We're given the vectors $\vec{v}_1 = (1,1)$ and $\vec{v}_2 = (0,1)$ and want to perform the Gram–Shmidt procedure. We pick $\vec{e}_1 = \vec{v}_1 = (1,1)$ and after normalization we have $\hat{e}_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. Next we compute $\vec{e}_2 = \vec{v}_2 - \Pi_{\hat{e}_1}(\vec{v}_2) = \vec{v}_2 - (\hat{e}_1 \cdot \vec{v}_2)\hat{e}_1 = (\frac{-1}{2}, \frac{1}{2})$. Normalizing $\vec{e}_2$ we obtain $\hat{e}_2 = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

P7.16 Let $\vec{v}_1 = (3,1)$ and $\vec{v}_2 = (-1,1)$. We start by identifying $\vec{e}_1 = \vec{v}_1$, then perform Gram–Shmidt process to find $\vec{e}_2$ from $\vec{v}_2$:

$$\vec{e}_2 = \vec{v}_2 - \Pi_{\hat{e}_1}(\vec{v}_2)\hat{e}_1 = \vec{v}_2 - \left(\frac{\vec{e}_1}{\|\vec{e}_1\|} \cdot \vec{v}_2\right)\frac{\vec{e}_1}{\|\vec{e}_1\|} = \left(\frac{-2}{5}, \frac{6}{5}\right).$$

Now we have two orthogonal vectors and we can normalize them to make them unit length. We obtain the vectors $(\frac{3}{\sqrt{10}}, \frac{1}{\sqrt{10}})$ and $(\frac{-1}{\sqrt{10}}, \frac{3}{\sqrt{10}})$ which form an orthogonal basis.

P7.17 First you have to find eigenvalues of the given matrix.1,1 Then find an eigenvector for each eigenvalue. Then construct a matrix Q composed of the 3 eigenvectors. Finally, write the eigendecomposition of the form $A = Q\Lambda Q^{-1}$.

P7.18 a) $\sqrt{3^2+4^2} = 5$; b) Only the sign of imaginary part changes so it becomes $2+3i$; c) $3i-1+3+2i = 2+5i$; d) $|3i-4i-5| = |-5-i| = \sqrt{26}$.

P7.21 First of all we must determine dimensionality of the vector space in question. The general vector space of 3×3 matrices has 9 dimensions, but a diagonal matrix A satisfies $a_{ij} = 0$ for all $i \neq j$, which corresponds to the following six constraints $\{a_{12}=0, a_{13}=0, a_{21}=0, a_{23}=0, a_{31}=0, a_{32}=0\}$. The vector space of diagonal matrices is therefore three dimensional. The answer given is the standard basis. Other answers are possible so long as they span the same space, and there are three of them.

P7.22 A matrix $A \in \mathbb{R}^{3\times 3}$ is symmetric if and only if $A^\mathsf{T} = A$. This means we can pick the entries on the diagonal arbitrarily, but the symmetry requirement leads to the constraints $a_{12} = a_{21}$, $a_{13} = a_{31}$, and $a_{23} = a_{32}$. Thus space of 3×3 symmetric matrices is six dimensional.

P7.23 A Hermitian matrix H is a matrix with complex coefficients that satisfies $H = H^\dagger$, or equivalently $h_{ij} = \overline{h_{ji}}$, for all i,j. A priori the space of 3×3 matrices with complex coefficients is 18 dimensional, for the real and imaginary parts of each of the nine coefficients. The Hermitian property imposes two types of constraints. Diagonal elements must be real if we want $h_{ii} = \overline{h_{ii}}$ to be true, which introduces three constraints. Once we pick the real and imaginary part of an off-diagonal element a_{ij}, we're forced to choose $a_{ji} = \overline{a_{ij}}$, which are another 6 constraints. Thus, the vector space of 3×3 Hermitian matrices is $18-3-6=9$ dimensional.

P7.24 To prove a matrix is nilpotent we can either compute its powers to see if we get the zero matrix. To prove a matrix is not nilpotent, you can show it has a non-zero eigenvalue.
(a) This matrix is nilpotent because its square is the zero matrix.
(b) The matrix is not nilpotent because its characteristic polynomial $p(\lambda) = \lambda^2 - 6\lambda + 8$ is different from the all-zero eigenvalues characteristic polynomial $p(\lambda) = (\lambda - 0)(\lambda - 0) = \lambda^2$.
(c) This matrix is nilpotent because it squares to the zero matrix.
(d) The matrix is not nilpotent because it has non-zero eigenvalues.
(e) Yes, the cube of this matrix is the zero matrix.
(f) Yes, the square of this matrix is the zero matrix.

P7.25 The characteristic polynomial of A is $p_A(\lambda) = \lambda^2 - 2\lambda = \lambda(\lambda - 2)$ so the eigenvalues are 0 and 2. The eigenvector for $\lambda = 0$ is $\vec{e}_0 = (-1 + i, 2)^\mathsf{T}$. The eigenvector for $\lambda = 2$ is $\vec{e}_2 = (1+i, 2)^\mathsf{T}$. As A has two one-dimensional eigenspaces, an eigenbasis is given by $\{(-1 + i, 2)^\mathsf{T}, (1 + i, 2)^\mathsf{T}\}$. So A is diagonalizable with $Q = \begin{bmatrix} -1+i & 1+i \\ 2 & 2 \end{bmatrix}$ and $\Lambda = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$.

P7.26 (a) Using linearity: $\langle \vec{v}_1, 2\vec{v}_2 + 3\vec{v}_3 \rangle = 2\langle \vec{v}_1, \vec{v}_2 \rangle + 3\langle \vec{v}_1, \vec{v}_3 \rangle = 6 + 3 = 9$.
(b) Using linearity in both entries: $\langle 2\vec{v}_1 - \vec{v}_2, \vec{v}_1 + \vec{v}_3 \rangle = 2\langle \vec{v}_1, \vec{v}_1 \rangle + 2\langle \vec{v}_1, \vec{v}_3 \rangle - \langle \vec{v}_2, \vec{v}_1 \rangle - \langle \vec{v}_2, \vec{v}_3 \rangle = 2\langle \vec{v}_1, \vec{v}_1 \rangle + 2\langle \vec{v}_1, \vec{v}_3 \rangle - \langle \vec{v}_1, \vec{v}_2 \rangle - \langle \vec{v}_2, \vec{v}_3 \rangle = 2 + 2 - 3 + 2 = 3$.
(c) We start with $13 = \langle \vec{v}_2, \vec{v}_1 + \vec{v}_2 \rangle = \langle \vec{v}_2, \vec{v}_1 \rangle + \langle \vec{v}_2, \vec{v}_2 \rangle$, and since we know $\langle \vec{v}_1, \vec{v}_2 \rangle = 3$, we obtain $3 + \|\vec{v}_2\|^2 = 13$, so $\|\vec{v}_2\|^2 = 10$ and $\|\vec{v}_2\| = \sqrt{10}$.

P7.27 (a) There are several ways to answer this question. One way is to note that the dimension of P_2 is 3 and so if B_a is a linearly independent set, then it is a basis as it has 3 elements. If $a(1 + ix) + b(1 + x + ix^2) + c(1 + 2ix) = 0$ then $(a + b + c)1 + (ia + b + 2ic)x + ibx^2 = 0$. But the standard basis of P_2 is linearly independent and so we must have $a + b + c = 0, ia + b + 2ic = 0$ and $ib = 0$. The last equation implies $b = 0$ and then the first two imply both a and c are zero. As the only linear combination of distinct elements of B_a which sums to zero is the trivial sum, B_a is linearly independent.

Chapter 8 solutions

Answers to exercises

E8.1 $4\text{Al} + 3\text{O}_2 \rightarrow 2\text{Al}_2\text{O}_3$. **E8.2** $\text{Fe(OH)}_3 + 3\text{HCl} \rightarrow \text{FeCl}_3 + 3\text{H}_2\text{O}$. **E8.3** **(a)** $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$; **(b)** $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$; **(c)** $A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$. **E8.4** **(a)** 1; **(b)** 1; **(c)** 2. **E8.5** $\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = \lambda_1 = \varphi = \frac{1+\sqrt{5}}{2}$. **E8.6** $a_N = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^N - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^N$. **E8.7** $S(m^*) = 4704.63$. **E8.8** $S(\vec{m}'^*) = 433.54$. **E8.9** $\vec{c} = \vec{x}G = (1, 0, 1, 0, 1, 0, 1)$. **E8.10** A 7×7 identity matrix next to an all-ones vectors.

Solutions to selected exercises

E8.3 The i^{th} row of the adjacency matrix contains the information of the outgoing edges for vertex i in the graph. If the edge (i, j) exists, then $A_{ij} = 1$, otherwise $A_{ij} = 0$.

E8.6 Another expression is $F_n = \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}}$.

E8.11 To find the inner product $\langle \mathbf{e}_1(x), \mathbf{e}_2(x) \rangle$, we must compute the integral $\int_0^L \sin(\frac{\pi}{L}x) \sin(\frac{2\pi}{L}x)\, dx$. We can change variables $y = \frac{\pi}{L}x$ to simplify the integral, changing also $dx \rightarrow \frac{L}{\pi}dy$ and the limits. We thus have $\langle \mathbf{e}_1(x), \mathbf{e}_2(x) \rangle =$

$k\int_0^\pi \sin(y)\sin(2y)\,dy$, where $k = \frac{L}{\pi}$. Using the double angle formula we find $\sin(y)\sin(2y) = \sin(y)2\sin(y)\cos(y) = 2\sin^2(y)\cos(y)$. We proceed using the substitution $u = \sin(y)$, $du = \cos(y)dy$ to obtain $2k\int \sin^2(y)\cos(y)\,dy = 2k\int u^2 du = \frac{2k}{3}\left[u^3\right] = \frac{2k}{3}\sin^3(y)$. Finally, we evaluating the expression at the endpoints: $\langle \mathbf{e}_1(x), \mathbf{e}_2(x)\rangle = \frac{2k}{3}\left[\sin^3(\pi) - \sin^3(0)\right] = 0$, confirming orthogonality.

E8.12 The coefficient b_0 corresponds to the basis function $\sin(\frac{0\pi}{T}t)$, which is zero everywhere. Thus, the integral $b_0 = \frac{1}{T}\int_0^T f(t)\sin(\frac{2\pi 0}{T}t)\,dt$, always produces a zero result $b_0 = 0$. On the other hand, the zero-frequency cos function is constant $\cos(\frac{2\pi 0}{T}t) = 1$, and the coefficient $a_0 = \frac{1}{T}\int_0^T f(t)1\,dt$ corresponds to the average value of the function $f(t)$ on the interval $[0, T]$.

Answers to problems

P8.1 **(a)** $A = \begin{bmatrix} 0&1&0&0&1\\ 1&0&1&1&0\\ 0&1&0&1&0\\ 0&0&1&0&1\\ 1&0&0&1&0 \end{bmatrix}$; **(b)** $A = \begin{bmatrix} 0&1&0&0&1\\ 0&0&1&0&1\\ 0&0&0&1&1\\ 1&0&0&0&1\\ 0&0&0&0&0 \end{bmatrix}$; **(c)** $A = \begin{bmatrix} 0&1&1&1&1\\ 1&0&1&0&0\\ 0&1&0&1&0\\ 0&0&1&0&1\\ 1&0&0&1&0 \end{bmatrix}$. **P8.2** **(a)** 1; **(b)** 1; **(c)** 3.

Solutions to selected problems

P8.1 The i^{th} row of the adjacency matrix contains the information of the outgoing edges for vertex i in the graph. If the edge (i, j) exists, then $A_{ij} = 1$, otherwise $A_{ij} = 0$.

P8.3 Decompose the formula for c_n into a its real part and imaginary parts:

$$\begin{aligned} c_n &= \int_0^T \left\langle e^{i\frac{2\pi n}{T}t}, \mathbf{e}_t \right\rangle f(t)\,dt \\ &= \frac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt \\ &= \text{Re}\left\{\tfrac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt\right\} + \text{Im}\left\{\tfrac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt\right\} i \\ &= \tfrac{1}{T}\int_0^T f(t)\text{Re}\left\{e^{-i\frac{2\pi n}{T}t}\right\}dt + \tfrac{1}{T}\int_0^T f(t)\text{Im}\left\{e^{-i\frac{2\pi n}{T}t}\right\} i\,dt \\ &= \tfrac{1}{T}\int_0^T f(t)\cos\left(\tfrac{2\pi n}{T}t\right)\,dt - \tfrac{1}{T}\int_0^T f(t)\sin\left(\tfrac{2\pi n}{T}t\right) i\,dt. \end{aligned}$$

We can recognize the real part of c_n as the cosine coefficients a_n of the Fourier series, and the imaginary part of c_n as the negative of the sine coefficients b_n of the Fourier series:

$$\text{Re}\{c_n\} = \text{Re}\left\{\tfrac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt\right\} = \tfrac{1}{T}\int_0^T f(t)\cos\left(\tfrac{2\pi n}{T}t\right)dt = a_n,$$

$$\text{Im}\{c_n\} = \text{Im}\left\{\tfrac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi n}{T}t}\,dt\right\} = \tfrac{1}{T}\int_0^T f(t)\sin\left(-\tfrac{2\pi n}{T}t\right)dt = -b_n.$$

Thus we have shown $c_n = a_n - ib_n$.

Using the simple definition of the coefficients c_n above, the synthesis equation for the complex Fourier transform is a somewhat awkward expression $f(t) = c_0 + \frac{1}{2}\sum_{n=1}^{\infty} e^{-i\frac{2\pi n}{T}t}c_n + \frac{1}{2}\sum_{n=-\infty}^{-1} e^{-i\frac{2\pi n}{T}t}\overline{c_n}$. Many textbooks describe complex Fourier series in terms of the two-sided coefficients $c'_n, n \in \mathbb{Z}$ defined as

$$\begin{aligned} c'_0 &= c_0, \\ c'_n &= \tfrac{1}{2}c_n = \tfrac{1}{2}(a_n - ib_n) && \text{for } n \geq 1, \\ c'_{-n} &= \tfrac{1}{2}\overline{c_n} = \tfrac{1}{2}(a_n + ib_n) && \text{for } n \geq 1. \end{aligned}$$

Using the coefficients c'_n, the synthesis equation for the complex Fourier series is the simpler expression $f(t) = \sum_{n=-\infty}^{\infty} e^{-i\frac{2\pi n}{T}t} c'_n$.

Chapter 9 solutions

Answers to exercises

E9.1 **(a)** No; weights don't add to one. **(b)** Yes. **(c)** No; contains a negative number. **E9.4** $p_{X_\infty} = \left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61}\right)$. **E9.5** $p_{X_\infty} = \left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61}\right)$. **E9.6** $p_{X_\infty} = (0.2793, 0.1457, 0.2768, 0.02, 0.2781)^\mathsf{T}$.

Solutions to selected exercises

E9.4 Use `ev = C.eigenvects()[0][2][0]` to extract the eigenvector that correspond to the eigenvalue $\lambda = 1$, then normalize the vector by its 1-norm to make it a probability distribution `pinf = ev/ev.norm(1) =` $\left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61}\right)$.

E9.5 Define the matrix `C` then use `ev = (C-eye(3)).nullspace()[0]` to obtain an eigenvector that corresponds the eigenvalue $\lambda = 1$. To obtain a probability distribution, compute `ev/ev.norm(1)` $= \left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61}\right)$. Using the `nullspace` method is a more targeted approach for finding stationary distributions than using the `eigenvects` method. We don't need to compute all the eigenvectors of C, just the one we're interested in.

E9.6 You can see the solution at this URL: `bit.ly/21GOUCe`.

Answers to problems

P9.1 $\Pr(\{\texttt{heads}, \texttt{heads}, \texttt{heads}\}) = \frac{1}{8}$. **P9.2** $p_{X_1} = (\frac{1}{2}, \frac{1}{2})^\mathsf{T}$, $p_{X_2} = (\frac{3}{8}, \frac{5}{8})^\mathsf{T}$, $p_{X_\infty} = (\frac{1}{3}, \frac{2}{3})^\mathsf{T}$.

Solutions to selected problems

P9.1 Substitute $p = \frac{1}{2}$ and $n = 3$ into the expression p^n.

P9.2 Defined X_i to be probability distribution of the weather in Year i. The transition matrix for the weather Markov chain is $M = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \frac{3}{4} \end{bmatrix}$. We obtain the weather in Year 1 using $p_{X_1} = M(1,0)^\mathsf{T} = (\frac{1}{2}, \frac{1}{2})^\mathsf{T}$. The weather in Year 2 is $p_{X_2} = M^2(1,0)^\mathsf{T} = (\frac{3}{8}, \frac{5}{8})^\mathsf{T}$. The long term (stationary) distribution is $p_{X_\infty} = M^\infty(1,0)^\mathsf{T} = (\frac{1}{3}, \frac{2}{3})^\mathsf{T}$.

Chapter 10 solutions

Answers to exercises

E10.1 $\vec{u} = (\alpha, 0, 0\beta)^\mathsf{T}$ (column vector), $\overline{\vec{v}^\mathsf{T}} = (0, a, b, 0)$ (row vector). **E10.2** $|w\rangle = 1|0\rangle + i|1\rangle - |2\rangle$, $\langle w| = 1\langle 0| - i\langle 1| - \langle 2|$, $\|\vec{w}\| = \sqrt{3}$. **E10.3** $\det(A) = -2$. **E10.6** See solution. **E10.7** $\Pr(\{-\}|\psi) = \frac{(\alpha-\beta)^2}{2}$.

Solutions to selected exercises

E10.5 Note XY is not the same as YX.

E10.6 We can rewrite the definition of $|+\rangle$ and $|-\rangle$ in order to obtain expressions for the elements of the standard basis $|0\rangle = \frac{1}{\sqrt{2}}|+\rangle + \frac{1}{\sqrt{2}}|-\rangle$, and $|1\rangle = \frac{1}{\sqrt{2}}|+\rangle - \frac{1}{\sqrt{2}}|-\rangle$. The remainder of the procedure is straightforward:

$$\begin{aligned}
|01\rangle - |10\rangle &= |0\rangle|1\rangle - |1\rangle|0\rangle \\
&= \Big(\tfrac{1}{\sqrt{2}}|+\rangle + \tfrac{1}{\sqrt{2}}|-\rangle\Big)\Big(\tfrac{1}{\sqrt{2}}|+\rangle - \tfrac{1}{\sqrt{2}}|-\rangle\Big) \\
&\qquad - \Big(\tfrac{1}{\sqrt{2}}|+\rangle - \tfrac{1}{\sqrt{2}}|-\rangle\Big)\Big(\tfrac{1}{\sqrt{2}}|+\rangle + \tfrac{1}{\sqrt{2}}|-\rangle\Big) \\
&= \frac{1}{2}\Big(\cancel{|+\rangle|+\rangle} - |+\rangle|-\rangle + |-\rangle|+\rangle - \cancel{|-\rangle|-\rangle} \\
&\qquad - \cancel{|+\rangle|+\rangle} - |+\rangle|-\rangle + |-\rangle|+\rangle + \cancel{|-\rangle|-\rangle}\Big) \\
&= \frac{1}{2}\Big(-2|+\rangle|-\rangle + 2|-\rangle|+\rangle\Big) \\
&= -(|+\rangle|-\rangle - |-\rangle|+\rangle) \\
&= |+\rangle|-\rangle - |-\rangle|+\rangle.
\end{aligned}$$

The last equality holds because the global phase of the states doesn't matter.

Answers to problems

P10.1 No, since Q is not unitary. **P10.2** $HH|0\rangle = |0\rangle$ and $HH|1\rangle = |1\rangle$. **P10.3** 2 parameters. **P10.4** $\alpha \in [0, 1]$ and $\varphi \in [0, 2\pi]$. **P10.5** $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$. **P10.6** $|R_3\rangle = |\psi\rangle_3$, since the state was teleported to Bob's register.

Solutions to selected problems

P10.1 Quantum operators are unitary. Since $QQ^\dagger \neq \mathbb{1}$, Q is not unitary and it cannot be implemented by any physical device. The boss is *not* always right!

P10.2 Let's first see what happens to $|0\rangle$ when we apply the operator HH. The result of the first H applied to $|0\rangle$ is $H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Then applying the second H operator we get $HH|0\rangle = H|+\rangle = \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle)$. Applying the H operation gives $\frac{1}{\sqrt{2}}\Big(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\Big)$, which simplifies to $\frac{1}{\sqrt{2}}\Big(\frac{2}{\sqrt{2}}|0\rangle\Big) = |0\rangle$. So $HH|0\rangle = |0\rangle$. A similar calculation shows $HH|1\rangle = |1\rangle$.

P10.3 Starting from the four degrees of freedom for a general two-dimensional complex vectors, we must subtract one degree of freedom for each of the constraints: one because we're ignoring global phase, and one because we require $|\alpha|^2 + |\beta|^2 = 1$:

$$4 \text{ d.f.} - \alpha \text{ real} - \{\||\psi\rangle\| = 1\} = 2 \text{ d.f.}$$

A qubit $|\psi\rangle$ has only two degrees of freedom. In other words two parameters are sufficient to describe any qubit.

P10.5 These are the angles on the *Bloch sphere*.

P10.6 Check this video for a sketch of the solution: `youtu.be/3wZ35c3oYUE`.

Appendix B

Notation

This appendix contains a summary of the notation used in this book.

Math notation

Expression	Read as	Used to denote
a, b, x, y		variables
$=$	is equal to	expressions that have the same value
$\equiv$	is defined as	new variable definitions
$a + b$	a plus b	the combined lengths of a and b
$a - b$	a minus b	the difference in lengths between a and b
$a \times b \equiv ab$	a times b	the area of a rectangle
$a^2 \equiv aa$	a squared	the area of a square of side length a
$a^3 \equiv aaa$	a cubed	the volume of a cube of side length a
a^n	a exponent n	a multiplied by itself n times
$\sqrt{a} \equiv a^{\frac{1}{2}}$	square root of a	the side length of a square of area a
$\sqrt[3]{a} \equiv a^{\frac{1}{3}}$	cube root of a	the side length of a cube with volume a
$a/b \equiv \frac{a}{b}$	a divided by b	a parts of a whole split into b parts
$a^{-1} \equiv \frac{1}{a}$	one over a	division by a
$f(x)$	f of x	the function f applied to input x
f^{-1}	f inverse	the inverse function of $f(x)$
$f \circ g$	f compose g	function composition; $f \circ g(x) \equiv f(g(x))$
e^x	e to the x	the exponential function base e
$\ln(x)$	natural log of x	the logarithm base e
a^x	a to the x	the exponential function base a
$\log_a(x)$	log base a of x	the logarithm base a
θ, ϕ	*theta*, *phi*	angles
$\sin, \cos, \tan$	sin, cos, tan	trigonometric ratios
$\%$	percent	proportions of a total; $a\% \equiv \frac{a}{100}$

Set notation

You don't need a lot of fancy notation to do math, but it really helps if you know a little bit of set notation.

Symbol	Read as	Denotes
$\{ \ldots \}$	the set ...	definition of a set
$\mid$	such that	describe or restrict the elements of a set
$\mathbb{N}$	the naturals	the set $\mathbb{N} \equiv \{0, 1, 2, \ldots\}$. Also $\mathbb{N}_+ \equiv \mathbb{N} \backslash \{0\}$.
$\mathbb{Z}$	the integers	the set $\mathbb{Z} \equiv \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$
$\mathbb{Q}$	the rationals	the set of fractions of integers
$\mathbb{R}$	the reals	the set of real numbers
$\mathbb{C}$		the set of complex numbers
$\mathbb{F}_q$	finite field	the set $\{0, 1, 2, 3, \ldots, q-1\}$
$\subset$	subset	one set strictly contained in another
$\subseteq$	subset or equal	containment or equality
$\cup$	union	the combined elements from two sets
$\cap$	intersection	the elements two sets have in common
$S \setminus T$	S set minus T	the elements of S that are not in T
$a \in S$	a in S	a is an element of set S
$a \notin S$	a not in S	a is not an element of set S
$\forall x$	for all x	a statement that holds for all x
$\exists x$	there exists x	an existence statement
$\nexists x$	there doesn't exist x	a non-existence statement

An example of a condensed math statement that uses set notation is "$\nexists m, n \in \mathbb{Z}$ such that $\frac{m}{n} = \sqrt{2}$," which reads "there don't exist integers m and n whose fraction equals $\sqrt{2}$." Since we identify the set of fraction of integers with the rationals, this statement is equivalent to the shorter "$\sqrt{2} \notin \mathbb{Q}$," which reads "$\sqrt{2}$ is irrational."

Vectors notation

Expression	Denotes
$\mathbb{R}^n$	the set of n-dimensional real vectors
$\vec{v}$	a vector
(v_x, v_y)	vector in component notation
$v_x\hat{\imath} + v_y\hat{\jmath}$	vector in unit vector notation
$\|\vec{v}\|\angle\theta$	vector in length-and-direction notation
$\|\vec{v}\|$	length of the vector $\vec{v}$
θ	angle the vector $\vec{v}$ makes with the x-axis
$\hat{v} \equiv \frac{\vec{v}}{\|\vec{v}\|}$	unit length vector in the same direction as $\vec{v}$
$\vec{u} \cdot \vec{v}$	dot product of the vectors $\vec{u}$ and $\vec{v}$
$\vec{u} \times \vec{v}$	cross product of the vectors $\vec{u}$ and $\vec{v}$

Complex numbers notation

Expression	Denotes
$\mathbb{C}$	the set of complex numbers $\mathbb{C} \equiv \{a + bi \mid a, b \in \mathbb{R}\}$
i	the unit imaginary number $i \equiv \sqrt{-1}$ or $i^2 = -1$
$\mathrm{Re}\{z\} = a$	real part of $z = a + bi$
$\mathrm{Im}\{z\} = b$	imaginary part of $z = a + bi$
$\lvert z\rvert\angle\varphi_z$	polar representation of $z = \lvert z\rvert \cos\varphi_z + i\lvert z\rvert \sin\varphi_z$
$\lvert z\rvert = \sqrt{a^2 + b^2}$	magnitude of $z = a + bi$
$\varphi_z = \tan^{-1}(b/a)$	phase or argument of $z = a + bi$
$\bar{z} = a - bi$	complex conjugate of $z = a + bi$
$\mathbb{C}^n$	the set of n-dimensional complex vectors

Vector space notation

Expression	Denotes
U, V, W	vector spaces
$W \subseteq V$	vector space W subspace of vector space V
$\{\vec{v} \in V \mid \langle \texttt{cond} \rangle\}$	subspace of vectors in V satisfying condition $\langle \texttt{cond} \rangle$
$\mathrm{span}\{\vec{v}_1, \ldots, \vec{v}_n\}$	span of vectors $\vec{v}_1, \ldots, \vec{v}_n$
$\dim(U)$	dimension of vector space U
$\mathcal{R}(M)$	row space of M
$\mathcal{N}(M)$	null space of M
$\mathcal{C}(M)$	column space of M
$\mathcal{N}(M^\mathsf{T})$	left null space of M
$\mathrm{rank}(M)$	rank of M; $\mathrm{rank}(M) \equiv \dim(\mathcal{R}(M)) = \dim(\mathcal{C}(M))$
$\mathrm{nullity}(M)$	nullity of M; $\mathrm{nullity}(M) \equiv \dim(\mathcal{N}(M))$
B_s	the standard basis
$\{\vec{e}_1, \ldots, \vec{e}_n\}$	an orthogonal basis
$\{\hat{e}_1, \ldots, \hat{e}_n\}$	an orthonormal basis
${}_{B'}[\mathbb{1}]_B$	the change-of-basis matrix from basis B to basis B'
Π_S	projection onto subspace S
$\Pi_{S^\perp}$	projection onto the orthogonal complement of S

Notation for matrices and matrix operations

Expression	Denotes
$\mathbb{R}^{m\times n}$	the set of $m \times n$ matrices with real coefficients
A	a matrix
a_{ij}	entry in the i^{th} row and j^{th} column of A
$\lvert A\rvert$	determinant of A, also denoted $\det(A)$
A^{-1}	matrix inverse
A^{T}	matrix transpose
$\mathbb{1}$	identity matrix; $\mathbb{1}A = A\mathbb{1} = A$ and $\mathbb{1}\vec{v} = \vec{v}$
AB	matrix-matrix product
$A\vec{v}$	matrix-vector product
$\vec{w}^{\mathsf{T}}A$	vector-matrix product
$\vec{u}^{\mathsf{T}}\vec{v}$	vector-vector inner product; $\vec{u}^{\mathsf{T}}\vec{v} \equiv \vec{u}\cdot\vec{v}$
$\vec{u}\vec{v}^{\mathsf{T}}$	vector-vector outer product
$\text{ref}(A)$	row echelon form of A
$\text{rref}(A)$	reduced row echelon form of A
$\text{rank}(A)$	rank of $A \equiv$ number of pivots in $\text{rref}(A)$
$A \sim A'$	matrix A' obtained from matrix A by row operations
$\mathcal{R}_1, \mathcal{R}_2, \ldots$	row operations, of which there are three types: $\rightarrow$ $R_i \leftarrow R_i + kR_j$: add k-times row j to row i $\rightarrow$ $R_i \leftrightarrow R_j$: swap rows i and j $\rightarrow$ $R_i \leftarrow mR_i$: multiply row i by constant m
$E_{\mathcal{R}}$	elementary matrix that corresponds $\mathcal{R}$; $\mathcal{R}(M) \equiv E_{\mathcal{R}}M$
$[\,A \mid \vec{b}\,]$	augmented matrix containing matrix A and vector $\vec{b}$
$[\,A \mid B\,]$	augmented matrix array containing matrices A and B
M_{ij}	minor associated with entry a_{ij}. See page 169.
$\text{adj}(A)$	adjugate matrix of A. See page 171.
$(A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}$	generalized inverse of A. See page 333.
$\mathbb{C}^{m\times n}$	the set of $m \times n$ matrices with complex coefficients
$A^{\dagger}$	Hermitian transpose; $A^{\dagger} \equiv (\overline{A})^{\mathsf{T}}$

Notation for linear transformations

Expression	Denotes
$T : \mathbb{R}^n \to \mathbb{R}^m$	linear transformation T (a vector function)
$M_T \in \mathbb{R}^{m \times n}$	matrix representation of T
$\text{Dom}(T) \equiv \mathbb{R}^n$	domain of T
$\text{CoDom}(T) \equiv \mathbb{R}^m$	codomain of T
$\text{Im}(T) \equiv \mathcal{C}(M_T)$	the image space of T
$\text{Ker}(T) \equiv \mathcal{N}(M_T)$	the kernel of T
$S \circ T(\vec{x})$	composition of linear transformations; $S \circ T(\vec{x}) \equiv S(T(\vec{x})) \equiv M_S M_T \vec{x}$
$M \in \mathbb{R}^{m \times n}$	an $m \times n$ matrix
$T_M : \mathbb{R}^n \to \mathbb{R}^m$	the linear transformation defined as $T_M(\vec{v}) \equiv M\vec{v}$
$T_{M^\mathsf{T}} : \mathbb{R}^m \to \mathbb{R}^n$	the adjoint linear transformation $T_{M^\mathsf{T}}(\vec{a}) \equiv \vec{a}^\mathsf{T} M$

Matrix decompositions

Expression	Denotes
$A \in \mathbb{R}^{n \times n}$	a matrix (assume diagonalizable)
$p_A(\lambda) \equiv \lvert A - \lambda \mathbb{1} \rvert$	characteristic polynomial of A
$\lambda_1, \ldots, \lambda_n$	eigenvalues of A = roots of $p_A(\lambda) \equiv \prod_{i=1}^n (\lambda - \lambda_i)$
$\Lambda \in \mathbb{R}^{n \times n}$	diagonal matrix of eigenvalues of A
$\vec{e}_{\lambda_1}, \ldots, \vec{e}_{\lambda_n}$	eigenvectors of A
$Q \in \mathbb{R}^{n \times n}$	matrix whose columns are eigenvectors of A
$A = Q\Lambda Q^{-1}$	eigendecomposition of A
$A = O\Lambda O^\mathsf{T}$	eigendecomposition of a normal matrix
$B \in \mathbb{R}^{m \times n}$	a generic matrix
$\sigma_1, \sigma_2, \ldots$	singular values of B
$\Sigma \in \mathbb{R}^{m \times n}$	matrix of singular values of B
$\vec{u}_1, \ldots, \vec{u}_m$	left singular vectors of B
$U \in \mathbb{R}^{m \times m}$	matrix of left singular vectors of B
$\vec{v}_1, \ldots, \vec{v}_n$	right singular vectors of B
$V \in \mathbb{R}^{n \times n}$	matrix of right singular vectors of B
$B = U\Sigma V^\mathsf{T}$	singular value decomposition of B

Abstract vector space notation

Expression	Denotes
$(V, F, +, \cdot)$	abstract vector space of vectors from the set V, whose coefficients are from the field F, addition operation "+" and scalar-multiplication operation "$\cdot$"
$\mathbf{u}, \mathbf{v}, \mathbf{w}$	abstract vectors
$\langle \mathbf{u}, \mathbf{v} \rangle$	inner product of vectors $\mathbf{u}$ and $\mathbf{v}$
$\|\mathbf{u}\|$	norm of $\mathbf{u}$
$d(\mathbf{u}, \mathbf{v})$	distance between $\mathbf{u}$ and $\mathbf{v}$